

**VŠB – Technická univerzita Ostrava**

**Fakulta strojní**

**Katedra automatizační techniky a řízení**

**Elektronické výukové pomůcky  
pro předmět Identifikace systémů**

**Electronic Learning Materials  
for the Subject of System Identification**

**Student:**

**Dendis Richard**

**Vedoucí bakalářské práce:**

**Ing. Škutová Jolana, Ph.D.**

**Ostrava 2013**

## Zadání bakalářské práce

Student: **Richard Dendis**  
Studijní program: B2341 Strojírenství  
Studijní obor: 3902R001 Aplikovaná informatika a řízení  
Téma: Elektronické výukové pomůcky pro předmět Identifikace systémů  
Electronic Learning Materials for the Subject of System Identification

### Zásady pro vypracování:

1. Zhodnoťte analytickou identifikaci systémů a popište metody numerické derivace a integrace.
2. Navrhněte koncept a vzhled elektronických podkladů pro předmět Identifikace systémů na základě předchozího seznámení s dostupnými elektronickými publikacemi.
3. Naplňte dílčí části výukových materiálů, navrhněte sadu příkladů pro řešení v programu MATLAB a sestavte nezbytné podklady zdrojových kódů.
4. Navrhněte scénáře a vytvořte animace v prostředí Adobe Captivate znázorňující postup řešení příkladů v programu MATLAB.
5. Zhodnoťte kriticky výsledky své bakalářské práce a navrhněte směr dalšího řešení.

### Seznam doporučené odborné literatury:

Noskievič, P. *Modelování a identifikace systémů*. Ostrava: Montanex, 2007. 276 stran. ISBN 80-7225-030-2.  
Noskievič, P. *Simulace systémů*. Ostrava: katedra automatizační techniky a řízení, VŠB-TU Ostrava, 1992. 217 stran. ISBN 80-7078-112-2.  
Navara, M. & Němček, A. *Numerické metody* [online]. Praha : ČVUT Praha, 2005 [cit. 2010-12-04]. Dostupné z [www: http://cmp.felk.cvut.cz/~navara/nm/NM.pdf](http://cmp.felk.cvut.cz/~navara/nm/NM.pdf)  
Kozák, Š. & Kajan, S. *Matlab-Simulink I*. Bratislava: STU v Bratislavě, 1999. ISBN 80-227-1213-2.  
Hoffman, J. D. *Numerical Methods for Engineers and Scientists*. CRC Press, 2001. 840 pages. ISBN 978-0824704438.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jolana Škutová, Ph.D.**

Datum zadání: 14.12.2012  
Datum odevzdání: 20.05.2013



prof. Ing. Jiří Tůma, CSc.  
vedoucí katedry



doc. Ing. Ivo Hlavatý, Ph.D.  
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem celou bakalářskou práci včetně příloh vypracoval samostatně pod vedením vedoucího bakalářské práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě .....*20.5.2013*.....

*Dušan*  
.....


podpis

Prohlašuji, že

- jsem byl seznámen s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo nevýdělečně ke své vnitřní potřebě bakalářskou práci užít (§ 35 odst. 3).
- souhlasím s tím, že bakalářská práce bude v elektronické podobě uložena v Ústřední knihovně VŠB-TUO k nahlédnutí a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že údaje o kvalifikační práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě ..... 10.5.2013

Dendis Richard  
Sv. Čecha 1132  
735 81, Bohumín

  
.....  
podpis

## ANOTACE BAKALÁŘSKÉ PRÁCE

DENDIS, R. *Elektronické výukové pomůcky pro předmět Identifikace systémů*. Ostrava: VŠB – Technická univerzita Ostrava, Fakulta strojní, Katedra automatizační techniky a řízení, 2013, 62 s. Vedoucí práce: Škutová, J.

Bakalářská práce se zabývá vytvořením elektronických výukových podkladů k podpoře výuky předmětu Identifikace systémů se zaměřením na analytickou identifikaci systémů, metody numerické integrace a metody numerické derivace. Kromě teoretických podkladů je důraz kladen na zpracování vzorových příkladů v programu MATLAB & Simulink s následnou interaktivní prezentací postupu řešení pomocí programu Adobe Captivate. Elektronická publikace je zpracována jako hypertextový dokument. Obsah dokumentu ve formě teoretických podkladů a řešených příkladů je dokumentován jako příloha bakalářské práce.

## ANNOTATION OF BACHELOR THESIS

DENDIS, R. *Electronic Learning Materials for the Subject of System Identification : Bachelor Thesis*. Ostrava: VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Control Systems and Instrumentation, 2013, 62 p. Thesis head: Škutová, J.

The bachelor thesis deals with the creation of electronic teaching materials to support teaching object of System identification, focusing on the analytical system identification, numerical integration methods and methods of numerical differentiation. Besides the theoretical materials the focus is on examples of sample processing in MATLAB & Simulink, followed by an interactive presentation solution procedure using by Adobe Captivate program. The electronic publication is prepared like a hypertext document. The content of the document in the form of theoretical materials and solved examples is documented like an attachment of bachelor thesis.

# Obsah

<b>Seznam použitého značení.....</b>	<b>7</b>
<b>Seznam použitých zkratk.....</b>	<b>8</b>
<b>Úvod.....</b>	<b>9</b>
<b>1 Numerické metody výpočtu integrálu .....</b>	<b>10</b>
1.1 Obdélníková metoda .....	11
1.2 Lichoběžníková metoda .....	12
1.3 Simpsonova metoda .....	13
<b>2 Numerické metody výpočtu diferenciálních rovnic .....</b>	<b>18</b>
2.1 Eulerova metoda .....	20
2.2 Metody Rungeova-Kuttova typu .....	20
2.3 Adamsovy-Bashforthovy metody .....	22
2.4 Adamsovy-Moultonovy metody .....	23
2.5 Metody prediktor-korektor .....	23
<b>3 Identifikace a modelování systémů.....</b>	<b>27</b>
3.1 Základní pojmy .....	28
3.2 Analytická identifikace systémů - popis systémů.....	29
3.3 Metoda postupné integrace .....	32
3.4 Metoda snižování řádu derivace .....	35
<b>4 Popis systému MATLAB® a nadstavby Simulink® .....</b>	<b>38</b>
4.1 MATLAB.....	38
4.2 Simulink.....	39
<b>5 Seznámení s programem Adobe Captivate.....</b>	<b>44</b>
5.1 Tvorba nového projektu .....	45
5.2 Editace projektu .....	47
5.3 Publikování projektu.....	48
<b>6 Návrh a tvorba elektronických výukových podkladů.....</b>	<b>49</b>
6.1 Elektronické publikace dostupné na stránkách katedry ATR.....	49
6.2 Tvorba v HTML editoru .....	51
6.3 Návrh vlastní elektronické publikace .....	52
<b>7 Závěr a zhodnocení dosažených výsledků.....</b>	<b>54</b>
<b>Seznam použité literatury.....</b>	<b>55</b>
<b>Seznam příloh .....</b>	<b>57</b>

## Seznam použitého značení

<b>A</b>	stavová matice systému
<b>B</b>	stavová matice řízení
<b>C</b>	výstupní matice systému
<b>D</b>	matice převodu
$E$	chyba (přesnost) výpočtu numerického integrálu
$E_L$	lokální chyba
$E_R$	chyba zaokrouhlovací
$E_T$	chyba metody
$G(s)$	obrazový přenos
$I$	Integrál; velikost plochy obrazce
$M$	horní odhad chyby
$P$	prediktor
$U(s)$	obraz vstupní veličiny
$Y(s)$	obraz výstupní veličiny
$a$	dolní mez intervalu
$a_i$	koeficient levé strany diferenciální rovnice; koeficient mnohočlenu ve jmenovateli přenosu
$b$	horní mez intervalu
$b_i$	koeficient pravé strany diferenciální rovnice; koeficient mnohočlenu v čitateli přenosu
$b_{kj}$	koeficient Adams-Bashforth-Moultonových metod

---

$c$	konstanta
$f$	funkce
$h$	krok řešení
$m$	počet kroků výpočtu integrálu; řád pravé strany diferenciální rovnice
$n$	počet uzlových bodů; řád levé strany diferenciální rovnice
$x_i$	ekvidistantní uzel
$u(t)$	vstupní (akční) veličina
$x(t)$	stavová veličina
$\dot{x}_t$	stavová proměnná
$y(t)$	výstupní veličina
$\varepsilon$	chyba akumulovaná
$\varphi$	interpolační polynom

## Seznam použitých zkratk

AVI	Audio Video Interleave, formát multimediálního souboru
DVD	Digital Video Disk
HTML	HyperText Markup Language, značkovací jazyk pro hypertext
MPEG	kompresní formát multimediálního souboru
PDF	Portable Document Format, formát dokumentu
SWF	soubor ve formátu vektorové grafiky
WYSIWYG	„what you see is what you get“, způsob tvorby webových stránek



## Úvod

S rozvojem informačních technologií roste i význam tvorby různých modelů, které mají za cíl co nejrealističtěji napodobit chování skutečných systémů. S tím roste význam i numerických metod. Přestože samotné numerické metody jsou známy již dlouho, jedním z důvodů jejich rychlého rozvoje je využívání výpočetní techniky při vytváření modelů.

Cílem této práce je vytvořit elektronické výukové podklady sloužící jako podpora výuky k předmětu Identifikace systémů, které by měly studenty seznámit s problematikou analytické identifikace systémů, se zaměřením na modelování systému pomocí blokových schémat v programu Simulink. Součástí výukových podkladů jsou také postupy řešení výpočtů metod numerické integrace a derivace v programu MATLAB. Postupy řešení vzorových příkladů jsou kromě textové statické formy prezentovány také dynamickými animacemi pomocí programu Adobe Captivate.

Úvodní část této práce je zaměřena na zpracování literární rešerše problematiky analytické identifikace systémů a metod numerické integrace a derivace. Součástí jednotlivých kapitol jsou matematická řešení vzorových příkladů. Následuje seznámení se s programy MATLAB & Simulink a Adobe Captivate, pomocí kterých je v této práci zpracováno řešení a prezentace vzorových příkladů. V závěrečné kapitole je popsán návrh a tvorba vzhledu a konceptu samotné elektronické publikace.

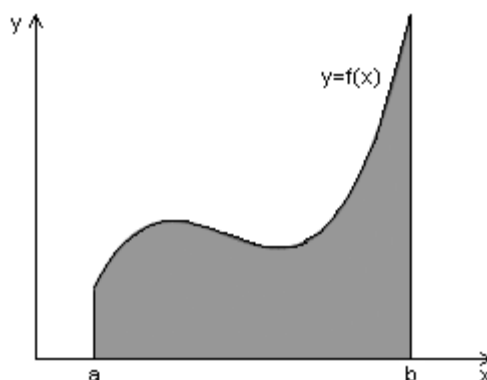
# 1 Numerické metody výpočtu integrálu

Součástí řešení různých úloh může být potřeba nalézt hodnotu určitého integrálu. Některé integrály lze vypočítat snadno pomocí tabulek a běžných integračních metod jako jsou substituce nebo per partes. V případě, že jsou hodnoty funkce dány tabulkou (např. výsledek předchozího měření nebo výpočtu), nelze u těchto integrálů použít standardní integrační metody, ale výpočet se provádí pomocí numerických metod.

Numerické metody výpočtu integrálu se zabývají výpočtem přibližné hodnoty určitého integrálu spojitě funkce  $f(x)$  na intervalu  $\langle a, b \rangle$

$$I = \int_a^b f(x) dx, \quad (1.1)$$

kde číslo  $I$  představuje z geometrického hlediska obsah plochy pod grafem funkce  $f(x)$  omezené na intervalu  $\langle a, b \rangle$ . Grafické znázornění integrálu je na obrázku 1.1.



Obrázek 1.1 Určitý integrál

Cílem použití numerických metod je výpočet velikosti plochy přibližného obrazce pomocí několika funkčních hodnot. Proto lze numerické metody použít pro jakoukoliv funkci  $f(x)$ , dokonce i pro funkci danou tabulkou. Pracnost výpočtu pomocí numerických metod je relativně malá.

Oblasti, kde je použití numerické integrace nezbytné nebo doporučené, jsou následující:

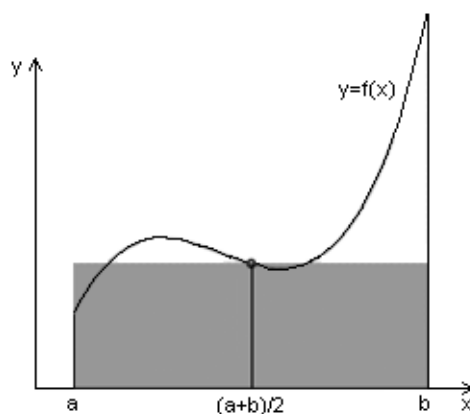
- Integrály nelze vypočítat analyticky.
- Analytické řešení určitého integrálu je výpočetně náročné.

- Integrovaná funkce není dána obecně matematickým vztahem, ale konkrétními hodnotami (např. tabulka nebo graf).
- Požadavek řešení integrálu numerickou integrací vyplývá z daného postupu řešení, např. při numerickém řešení diferenciálních rovnic.

Postup řešení výpočtu integrálu vzorového příkladu bude proveden pomocí následujících numerických metod.

## 1.1 Obdélníková metoda

Obdélníková metoda se řadí mezi metody s otevřeným vzorcem, to znamená, že krajní body intervalu nejsou uzly kvadratury. Uzlem interpolace uvažujeme střed intervalu  $\langle a, b \rangle$ . Funkce na tomto intervalu se nahradí konstantou  $f\left(\frac{a+b}{2}\right)$  a výsledná přibližná hodnota integrálu je rovna obsahu obdélníka, jak je znázorněno na obrázku 1.2.



Obrázek 1.2 Určitý integrál - obdélníková metoda

Položíme-li  $n = 0$ ,  $x_0 = \frac{a+b}{2}$  a jako interpolační polynom uvažujeme konstantní funkci  $\varphi(x) = f\left(\frac{a+b}{2}\right)$  poté

$$\int_a^b f(x) dx \approx (b-a) \cdot f\left(\frac{a+b}{2}\right) = I_{Obd}. \quad (1.2)$$

Složený vzorec s počtem kroků  $m$  pro výpočet určitého integrálu pomocí obdélníkové metody má následující tvar

$$I_{Sobd} \approx \frac{(b-a)}{m} \cdot \sum_{i=1}^m f\left(\frac{x_{i-1} + x_i}{2}\right). \quad (1.3)$$

Odhad chyby výpočtu integrálu pomocí obdélníkové metody je dán vztahem

$$|E(f)| \leq \frac{1}{2}(b-a)^2 \cdot M_1, \quad (1.4)$$

kde

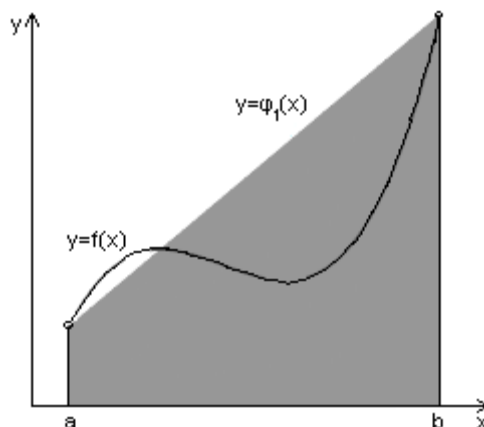
$$M_1 = \max_{t \in \langle a, b \rangle} |f^{(1)}(t)|. \quad (1.5)$$

## 1.2 Lichoběžníková metoda

Lichoběžníková metoda je nejjednodušší metodou ze skupiny uzavřených vzorců, u nichž oba krajní body intervalu jsou uzly kvadratury. Odhad určitého integrálu funkce  $f(x)$ , na intervalu  $\langle a, b \rangle$ , je dán vztahem

$$\int_a^b f(x) dx \approx \frac{(b-a)}{2} \cdot [f(a) + f(b)] = I_{Lich}. \quad (1.6)$$

Výsledná přibližná hodnota integrálu je rovna obsahu příslušného lichoběžníku, jehož rovnoběžné strany mají délky  $f(a)$ ,  $f(b)$  a jehož výška je rovna délce intervalu  $\langle a, b \rangle$ , tzn.  $(b-a)$ , viz obrázek 1.3.



Obrázek 1.3 Určitý integrál - lichoběžníková metoda

Složený vzorec pro výpočet určitého integrálu pomocí lichoběžníkové metody má tvar

$$I_{SLich} \approx \frac{(b-a)}{m} \cdot \left[ \frac{1}{2} f(x_0) + \sum_{i=1}^{m-1} f(x_i) + \frac{1}{2} f(x_m) \right]. \quad (1.7)$$

Odhad chyby výpočtu integrálu pomocí lichoběžníkové metody je dán vztahem

$$|E(f)| \leq \frac{1}{12}(b-a)^3 \cdot M_2, \quad (1.8)$$

kde

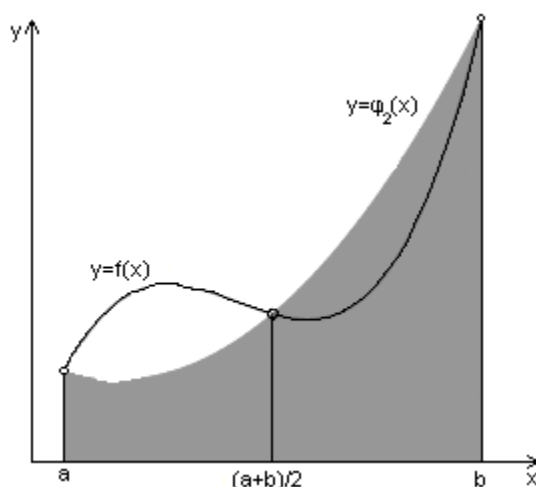
$$M_2 = \max_{t \in \langle a, b \rangle} |f''(t)|. \quad (1.9)$$

### 1.3 Simpsonova metoda

Simpsonova metoda je založena na integraci interpolačního polynomu druhého stupně. Za jeho uzly považujeme krajní body a střed intervalu  $\langle a, b \rangle$

$$\int_a^b f(x) dx \approx \frac{(b-a)}{6} \cdot \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] = I_{Simps}. \quad (1.10)$$

Interpolačním polynomem je kvadratická funkce, která je znázorněna na obrázku 1.4.



Obrázek 1.4 Určitý integrál - Simpsonova metoda

V případě složeného Simpsonova pravidla rozdělíme interval  $\langle a, b \rangle$  na sudý počet  $2m$  stejně dlouhých dílků pro  $m \geq 2$  s krokem  $h = (b-a)/2m$ . Z toho vyplývá lichý počet uzlů  $x_i = a + ih$ , kde  $i = 0, 1, \dots, 2m$ .

Vztah pro výpočet složeného Simpsonova pravidla je

$$I_{SSimps} \approx \frac{(b-a)}{6} \cdot \left[ f(x_0) + 4 \sum_{i=1}^m f(x_{2i-1}) + 2 \sum_{i=1}^{m-1} f(x_{2i}) + f(x_{2m}) \right]. \quad (1.11)$$

Odhad chyby výpočtu integrálu pomocí Simpsonovy metody je dán vztahem

$$|E(f)| \leq \frac{1}{90} \left( \frac{b-a}{2} \right)^5 \cdot M_4, \quad (1.12)$$

kde

$$M_4 = \max_{t \in \langle a, b \rangle} |f^{(4)}(t)|. \quad (1.13)$$

### **Příklad 1.1**

Pomocí obdélníkového, lichoběžníkového a Simpsonova pravidla vypočtete přibližnou hodnotu integrálu  $\int_1^4 \sqrt{x} dx$  pro počet dílků  $m = 1$  a určete odhadovanou chybu výpočtu metod numerické integrace, kde  $M_1 = \frac{1}{2}$ ,  $M_2 = \frac{1}{4}$  a  $M_3 = \frac{15}{16}$ .

### **Řešení**

$$I = \int_1^4 \sqrt{x} dx = \frac{2}{3} \left[ x^{\frac{3}{2}} \right]_1^4 = \frac{2}{3} \left( 4^{\frac{3}{2}} - 1^{\frac{3}{2}} \right) = \frac{2}{3} (8 - 1) = \frac{14}{3} = 4,6666667$$

$$I_{Obl} \approx (b-a) \cdot f\left(\frac{a+b}{2}\right) = (4-1) \cdot \sqrt{\frac{(4+1)}{2}} = 4.7434164902$$

$$|E(f)| \leq \frac{1}{2} (b-a)^2 \cdot M_1 = \frac{1}{2} (4-1)^2 \cdot \frac{1}{2} = 2,25$$

$$I_{Lich} \approx \frac{(b-a)}{2} \cdot [f(a) + f(b)] = \frac{(4-1)}{2} (\sqrt{1} + \sqrt{4}) = \frac{9}{2} = 4,5$$

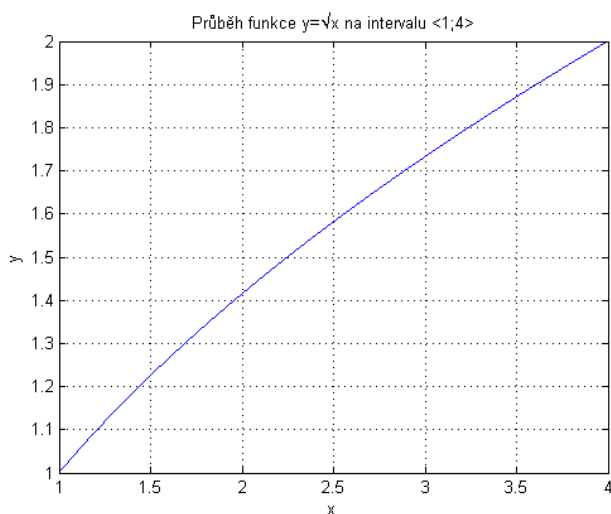
$$|E(f)| \leq \frac{1}{12} (b-a)^3 \cdot M_2 = \frac{1}{12} (4-1)^3 \cdot \frac{1}{4} = 0,5625$$

$$I_{Simps} \approx \frac{(b-a)}{6} \cdot \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] = \frac{(4-1)}{6} \left( \sqrt{1} + 4 \cdot \sqrt{\frac{(4+1)}{2}} + \sqrt{4} \right)$$

$$I_{Simps} \approx 4.6666666487$$

$$|E(f)| \leq \frac{1}{90} \left( \frac{b-a}{2} \right)^5 \cdot M_4 = \frac{1}{90} \left( \frac{4-1}{2} \right)^5 \cdot \frac{15}{16} = 0,079101563$$

Algoritmus výpočtu integrálu je obsažen v souborech **int\_obd.m**, **int\_lich.m** a **int\_simp.m**.



**Graf 1.1** Průběh funkce  $y = \sqrt{x}$  na intervalu  $\langle 1;4 \rangle$

Přehledné porovnání přibližných hodnot integrálů a přesnosti výpočtů jednotlivých metod je uvedeno v tabulce 1.1.

**Tabulka 1.1** Porovnání výsledků výpočtů integrálu a chyby výpočtu

	Hodnota integrálu	Chyba výpočtu integrálu
Obdélníková metoda	4,7434164902	2,2500
Lichoběžníková metoda	4.5	0,5625
Simpsonova metoda	4,6666666487	0,0791
Analytický výpočet	4,6666667	-

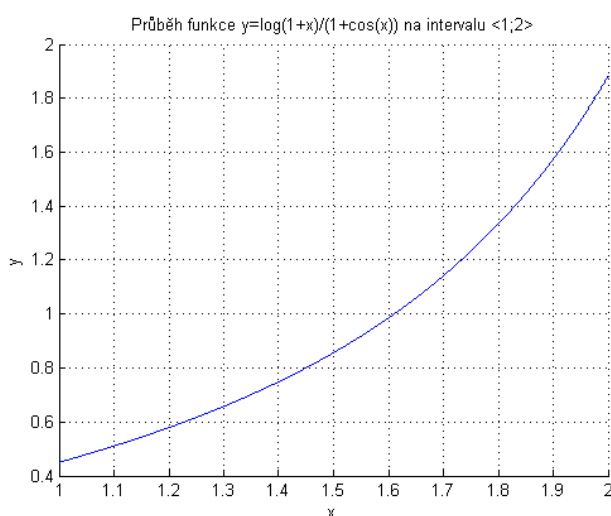
**Příklad 1.2**

Vypočtete hodnotu integrálu  $\int_1^2 \frac{\ln(1+x)}{1+\cos(x)} dx$  pomocí lichoběžníkové metody s přesností  $E = 10^{-6}$ .

**Řešení**

Výsledek integrálu se zadanou přesností  $E = 10^{-6}$  je

$$\int_1^2 \frac{\ln(1+x)}{1+\cos(x)} dx = 0.952187 \pm 0.000001 \quad \text{pro } n=1024$$



**Graf 1.2** Průběh funkce  $y=\log(1+x)/(1+\cos(x))$  na intervalu  $\langle 1;2 \rangle$

Integrál je vypočten pomocí algoritmu obsaženého v souboru `int_lich_presnost.m`.

**Příklad 1.3**

Aplikujte obdélníkovou, lichoběžníkovou a Simpsonovu metodu pro numerický výpočet určitého integrálu  $\int_0^\pi \sin(x) dx$  pro krok  $h = 2, 4, 6, \dots, 18, 20$  a porovnejte je s hodnotou integrálu stanovenou matematickou integrací goniometrické funkce na daném intervalu.

**Řešení**

Provedeme výpočet přesné hodnoty integrálu goniometrické funkce  $\int_0^\pi \sin(x) dx$ .

$$I = \int_0^\pi \sin(x) dx = [-\cos(x)]_0^\pi = -\cos(\pi) - (-\cos(0)) = -(-1) - (-1) = 2$$

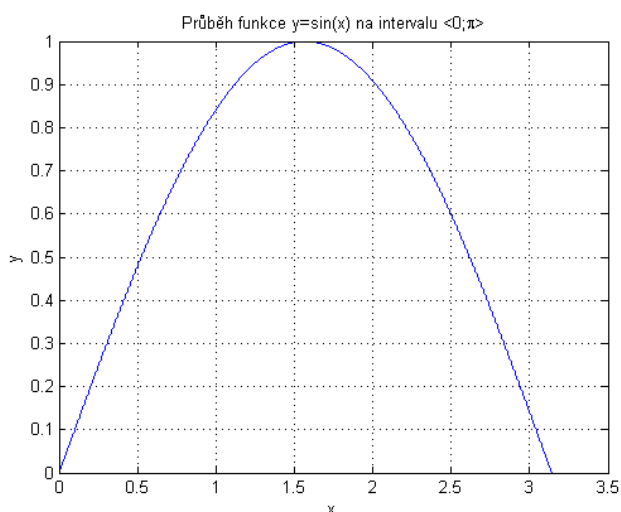


Pro jednotlivé  $h$  provedeme výpočet přibližných hodnot integrálu a výsledné hodnoty porovnáme s přesnou hodnotou. Výsledné hodnoty integrálu a porovnání jejich přesnosti je přehledně uvedeno v tabulce 1.2.

**Tabulka 1.2 Přibližné hodnoty integrálu a jejich přesnosti**

$h$	$I_{\text{obd}}$	$E_{\text{obd}} =  I - I_{\text{obd}} $	$I_{\text{lich}}$	$E_{\text{lich}} =  I - I_{\text{lich}} $	$I_{\text{simpson}}$	$E_{\text{simpson}} =  I - I_{\text{simpson}} $	$I_{\text{simp}} \text{ (QUAD)}$	$E_{\text{simp}} =  I - I_{\text{simp}}  \text{ (QUAD)}$
2	2,221441	0,221441	1,570796	0,429204	2,094395	0,094395	1,9999999964	0,0000000036
4	2,052344	0,052344	1,896119	0,103881	2,004560	0,004560		
6	2,023030	0,023030	1,954097	0,045903	2,000863	0,000863		
8	2,012909	0,012909	1,974232	0,025768	2,000269	0,000269		
10	2,008248	0,008248	1,983524	0,016476	2,000110	0,000110		
12	2,005723	0,005723	1,988564	0,011436	2,000053	0,000053		
14	2,004202	0,004202	1,991600	0,008400	2,000028	0,000028		
16	2,003216	0,003216	1,993570	0,006430	2,000017	0,000017		
18	2,002541	0,002541	1,994920	0,005080	2,000010	0,000010		
20	2,002058	0,002058	1,995886	0,004114	2,000007	0,000007		

Z výsledných hodnot je patrné, že obdélníková a lichoběžníková metoda určí hodnotu integrálu z hlediska přesnosti stejně, pouze aplikací Simpsonovy metody lze získat přesnější výsledky o několik řádů. Pro názornost uvádím přibližnou hodnotu integrálu získanou pomocí funkce QUAD, které počítá s přesností  $10^{-6}$ .



**Graf 1.3 Průběh funkce  $y=\sin(x)$  na intervalu  $\langle 0;\pi \rangle$**

Výpočty integrálu jsem provedl pomocí algoritmů obsažených v souborech `int_obd.m`, `int_lich.m`, `int_simp.m` a `int_simp_Q`.

## 2 Numerické metody výpočtu diferenciálních rovnic

Při popisu praktických úloh se vyskytuje jen zlomek diferenciálních rovnic, které lze řešit exaktně. Přestože lze sestavit vzorce popisující analytické řešení, bývá toto řešení velmi složité a obtížně lze s nimi dále pracovat. Proto znalost principů řešení diferenciálních rovnic pomocí numerických metod je základem k pochopení metod číslkové simulace dynamických systémů.

Princip řešení výpočtu diferenciální rovnice pomocí numerických metod lze demonstrovat na obyčejné diferenciální rovnici prvního řádu

$$y' = f(x, y), \quad (2.1)$$

s počáteční podmínkou  $y_0 = y(x_0)$  na intervalu  $\langle a, b \rangle$  (Cauchyova úloha), kde  $a = x_0$  a  $b = x_n$ .

Numerickým řešením je posloupnost hodnot

$$\{y(x_i)\} = y(x_0), y(x_1), \dots, y(x_n), \quad (2.2)$$

kde  $x_i$  je z intervalu řešení navýšený o krok  $h$  oproti předcházejícímu uzlu.

Krok  $h$  je roven rozdílu dvou sousedních hodnot nezávisle proměnné  $x$  dle vztahu

$$h_i = x_{i+1} - x_i, \quad i = 0, 1, \dots, n-1, \quad (2.3)$$

nebo je roven podílu z rozdílu krajních hodnot intervalu  $\langle a, b \rangle$  a počtu uzlových bodů řešení  $n$ , tedy

$$h = \frac{(b - a)}{n}. \quad (2.4)$$

Postup výpočtu probíhá cyklicky v krocích. Z hodnot známého řešení  $y_i$  v uzlu  $x_i$  se počítá řešení  $y_{i+1}$  v uzlu  $x_{i+1}$ . Krok numerického řešení zůstává mezi jednotlivými uzly konstantní. Změna délky kroku řešení závisí na velikosti chyby numerického řešení nebo na stabilitě výpočtu. [NOSKIEVIČ, 1992]

Každý uzel  $x_i$  je při numerickém řešení diferenciální rovnice zatížen chybou. V případě výsledné chyby hovoříme o tzv. lokální chybě  $E_L$ . Lokální chyba má dvě složky

- chybu metody  $E_T$ ,
- chybu zaokrouhlovací  $E_R$ .

Vzájemný vztah těchto chyb má tvar

$$E_L = E_T + E_R. \quad (2.5)$$

Chyba metody  $E_T$ , neboli zbytková chyba je udávána řádově v porovnání s mocninou kroku integrace  $h$ .

Chyba zaokrouhlovací  $E_R$  je určena počtem desetinných míst použitých při výpočtech.

Chyba akumulovaná  $\varepsilon$  charakterizuje nepřesnost výsledku po  $i$  krocích. Chyba výsledku jednoho kroku ovlivňuje řešení v krocích následujících

$$\varepsilon_i = Y(x_i) - y(x_i), \quad (2.6)$$

kde  $Y(x_i)$  je přesné řešení v bodě  $x_i$  a  $y(x_i)$  je numerickým řešením posloupnosti hodnot.

Numerické metody pro řešení diferenciálních rovnic rozdělujeme do dvou základních skupin, na jednokrokové a vícekové metody.

### Jednokrokové metody

Při řešení diferenciálních rovnic pomocí jednokrokových metod se při výpočtu řešení  $y_{i+1}$  vychází z již známé hodnoty řešení v předchozím uzlu  $y_i$ , nebo hodnot jiných bodů mezi oběma uzly  $y_i$  a  $y_{i+1}$ . Obecně lze řešení jednokrokových metod zapsat ve tvaru

$$y_{i+1} = y_i + h \cdot \varphi(x_i, y_i, h), \quad (2.7)$$

kde  $h \cdot \varphi(x_i, y_i, h)$  vyjadřuje přírůstek řešení mezi jednotlivými uzly  $i$  a  $i+1$ .

Přírůstek hledaného řešení se vyjadřuje z Taylorova rozvoje řadou podle mocnin  $h = x_{i+1} - x_i$  následovně

$$y_{i+1} - y_i = h \cdot f(x_i, y_i) + \frac{h^2}{2!} \cdot f'(x_i, y_i) + \dots + \frac{h^n}{n!} \cdot f^{(n-1)}(x_i, y_i) + 0 \cdot (h^{(n+1)}). \quad (2.8)$$

## Vícekrokové metody

Vícekrokové metody využívají při svém řešení i výsledky předcházejících kroků.

Daná metoda se nazývá *k-kroková*, jestliže při výpočtu  $y_{i+1}$  je potřeba dosadit do vzorce metody hodnoty řešení z  $k$  předchozích kroků, tj. hodnoty  $y_i, y_{i-1}, \dots, y_{i-k+1}$ . [KUČERA, 2007]

Komplikací při řešení úloh pomocí *k-krokových* metod je to, že při zahájení výpočtu je nutno znát hodnoty předcházejících  $k$  uzlů  $y_0, y_1, \dots, y_{k-1}$ . Jejich výpočet je nutno provést vhodnou jedнокrokovou (startovací) metodou a to stejného nebo vyššího řádu.

### 2.1 Eulerova metoda

Eulerova metoda využívá ke stanovení přírůstku pouze první člen řady Taylorova rozvoje, který obsahuje pouze funkci  $f(x_i, y_i)$  definovanou přímo diferenciální rovnicí

$$y_{i+1} = y_i + h \cdot f(x_i, y_i). \quad (2.9)$$

Zbytková chyba Eulerovy metody je určena zbývajících (zanedbanými) členy Taylorova rozvoje

$$E_T = \frac{h^2}{2!} \cdot f'(x_i, y_i) + \dots + \frac{h^n}{n!} \cdot f^{(n-1)}(x_i, y_i) + O(h^{(n+1)}). \quad (2.10)$$

Zbytková chyba Eulerovy metody dosahuje velkých hodnot a není tedy vhodná pro přesné výpočty diferenciálních rovnic.

Výhodou Eulerovy metody je její jednoduchost a rychlost řešení. Její využití spočívá zejména v předběžné číslíkové simulaci, kdy hlavním úkolem je pouze orientačně vyšetřit chování simulovaného systému.

### 2.2 Metody Rungeova-Kuttova typu

Metody Rungeova-Kuttova typu vychází z Taylorova rozvoje a berou v potaz oproti Eulerově metodě i členy vyšších řádů. Potřebné derivace funkce  $f(x, y)$  se počítají složitější diferenční metodou pomocí dalších pomocných bodů (tzv. přírůstků) mezi sousedními uzly v síti.

**Rungeova-Kuttova metoda 2. řádu (Heunova metoda)**

Heunova metoda je zobecněním lichoběžníkové metody integrace. Při řešení se nejprve vypočtou pomocné body a hodnoty derivace v nich a následně se nahradí integrál jejich lineární kombinací

$$y_0 = c, \quad (2.11)$$

$$k_1 = f(x_i, y_i), \quad (2.12)$$

$$k_2 = f(x_{i+1}, y_i + h_i k_1), \quad (2.13)$$

$$y_{i+1} = y_i + \frac{1}{2} h_i (k_1 + k_2). \quad (2.14)$$

**Rungeova-Kuttova metoda 4. řádu (RK4)**

Metoda RK4 je zobecněním Simpsonovy metody. Při řešení se nejprve vypočítají pomocné body a hodnoty derivace v nich a následně se nahradí integrál jejich lineární kombinací

$$y_0 = c, \quad (2.15)$$

$$k_1 = f(x_i, y_i), \quad (2.16)$$

$$k_2 = f\left(x_i + \frac{1}{2} h_i, y_i + \frac{1}{2} h_i k_1\right), \quad (2.17)$$

$$k_3 = f\left(x_i + \frac{1}{2} h_i, y_i + \frac{1}{2} h_i k_2\right), \quad (2.18)$$

$$k_4 = f(x_{i+1}, y_i + h_i k_3), \quad (2.19)$$

$$y_{i+1} = y_i + \frac{1}{6} h_i (k_1 + 2k_2 + 2k_3 + k_4). \quad (2.20)$$

Výhodou Rungeových-Kuttových metod je jednoduchá změna délky kroku řešení, které lze průběžně měnit v závislosti na zjištěné chybě řešení. Oproti Eulerově metodě lze dosáhnout srovnatelné přesnosti výsledků při daleko větším kroku  $h$ , tzn. při výrazně menším celkovém objemu výpočtů.

Nevýhodou může být pracnost výpočtu složitých funkcí. Výpočet se provádí tolikrát, kolikátého řádu je použitá metoda. V případě použití sofistikovaných programů se tato nevýhoda metody jeví jako nepodstatná.

V případě složitějších funkcí, je-li jejich výpočet pracný, je vhodné použít Rungeovy-Kuttovy metody pouze k výpočtu počátečních hodnot a následně použít některou z vícekových metod.

### 2.3 Adamsovy-Bashforthovy metody

Adamsovy-Bashforthovy metody jsou prediktorové metody numerického řešení diferenciálních rovnic, které vycházejí z explicitních vícekových metod.

Každou  $k+1$  krokovou explicitní Adamsovu-Bashforthovu metodu lze zapsat v obecném tvaru

$$y_{i+1} = y_i + h \sum_{j=0}^k b_{kj} f_{i-j}. \quad (2.21)$$

Hodnoty koeficientů  $b_{kj}$  explicitních metod pro  $k = 0, 1, 2, 3$  jsou uvedeny v tabulce 2.1.

**Tabulka 2.1** Hodnoty koeficientů Adamsovy-Bashforthovy metody

j	0	1	2	3
$b_{0j}$	1			
$b_{1j}$	$\frac{3}{2}$	$-\frac{1}{2}$		
$b_{2j}$	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$	
$b_{3j}$	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$

[NOSKIEVIČ, 1992]

## 2.4 Adamsovy-Moultonovy metody

Adamsovy-Moultonovy metody jsou korektorové metody numerického řešení diferenciálních rovnic, které vycházejí z implicitních vícekrokových metod.

Každou  $k$ -krokovou implicitní Adamsovu-Moultonovu metodu lze zapsat v obecném tvaru

$$y_{i+1} = y_i + h \sum_{j=0}^k b_{kj} f_{i+1-j}. \quad (2.22)$$

Hodnoty koeficientů  $b_{kj}$  implicitních metod pro  $k = 0, 1, 2, 3$  jsou uvedeny v tabulce 2.2.

**Tabulka 2.2** Hodnoty koeficientů Adamsovy-Moultonovy metody

j	0	1	2	3
$b_{0j}$	1			
$b_{1j}$	$\frac{1}{2}$	$\frac{1}{2}$		
$b_{2j}$	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$	
$b_{3j}$	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$

[NOSKIEVIČ, 1992]

## 2.5 Metody prediktor-korektor

Metoda prediktor-korektor vzniká kombinací obou typů vícekrokových metod.

Účinnost metody závisí mimo jiné na prvním odhadu  $y_{i+1}^0$ . Algoritmus prediktor-korektor používá pro první odhad Adamsovu-Bashforthovu metodu. Je žádoucí, aby byla použita metoda stejného řádu jako následně použitá implicitní metoda. První odhad explicitní metodou se nazývá prediktor (P). Jeden krok iterace se skládá z výpočtu funkce  $f(x)$  a následné opravy nového stavu, který se nazývá korektor (C).

Startovací hodnoty se vypočtou pomocí některé z jednokrokových metod.

Pro  $k$  iterací se metoda obecně značí  $P(EC)^k$ . Jako další označení algoritmu lze používat  $PE(CE)^k$ ,  $P(EC)^kE$ . V případě jediného průchodu korektorem se používá algoritmus PECE.

Zbytková chyba metody prediktor-korektor se provádí porovnáním hodnot prediktoru a korektoru

$$E_T(x_{i+1}) \approx |y_{i+1}^P - y_{i+1}^C|. \quad (2.23)$$

Je-li hodnota chyby  $E_T$  menší než stanovená mez, považuje se za hodnotu řešení v uzlu  $x_{i+1}$  hodnota  $y_i = y_{i+1}^C$ . V opačném případě se hodnota považuje za novou hodnotu prediktoru a opětovným použitím pouze korektorového vztahu dojde k zpřesnění řešení. Pokračuje se v testování velikosti chyby  $E_T$ .

### **Příklad 2.1**

Pomocí Eulerovy metody, metod Rungeova-Kuttova typy a Adamsovy-Bashforthovy metody řešte diferenciální rovnici  $y' = 0,2 \cdot y + \sin(x)$  na intervalu  $\langle -5, 5 \rangle$  s krokem  $h = 1$  a počáteční podmínkou  $y(0) = 0$ .

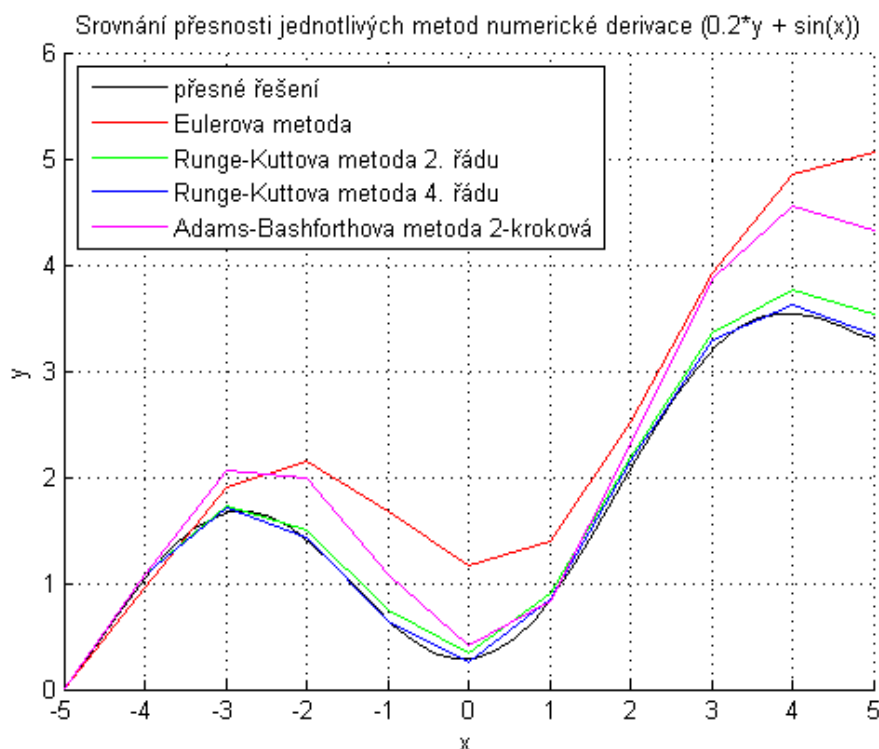
### **Numerické řešení výpočtu diferenciální rovnice**

Hodnoty numerického řešení diferenciální rovnice jsou zpracovány formou tabulky a graficky prezentovány pomocí grafů.

**Tabulka 2.3** Výsledky početního řešení jednotlivých numerických metod

i	$x_i$	Eulerova metoda	Rungeova-Kuttova metoda		Adamsova-Bashforthova metoda (2-kroková)
			2. řádu	4. řádu	
0	-5	0	0	0	0
1	-4	0.9589	1.0734	1.0860	1.0860
2	-3	1.9075	1.7360	1.7175	2.0676
3	-2	2.1479	1.5054	1.4346	1.9892
4	-1	1.6682	0.7481	0.6453	1.0858
5	0	1.1603	0.3492	0.2557	0.4050
6	1	1.3924	0.9054	0.8446	0.8387
7	2	2.5124	2.1862	2.1404	2.3120
8	3	3.9241	3.3566	3.2812	3.8649
9	4	4.8501	3.7584	3.6210	4.5502
10	5	5.0633	3.5320	3.3398	4.3230





Graf 2.1 Grafické řešení srovnání přesnosti jednotlivých metod numerické derivace

Algoritmy výpočtů diferenciálních rovnic jsou obsaženy v souborech **EulerovaMetoda.m**, **RungeKutt2radu.m**, **RungeKutt4radu.m** a **AdamsBashf2krok.m**.

### Příklad 2.2

Pomocí Adams-Bashforthových vícekrokových metod vyšetřete průběh diferenciální rovnice  $y' = x^2 - 0,2y$  na intervalu  $\langle -2, 3 \rangle$  s krokem  $h = 0,5$  a počáteční podmínkou  $y(0) = -1$ .

### Numerické řešení výpočtu diferenciální rovnice

Nejprve je nutno vypočítat výsledky řešení  $y_i$ , pro  $k$  předchozích kroků, tj. hodnoty  $y_i$ ,  $y_{i-1}$  a  $y_{i-2}$ . K výpočtu byla použita Rungeova-Kuttova metoda 4. řádu.

Tabulka 2.4 Výpočet hodnot k-předchozích kroků

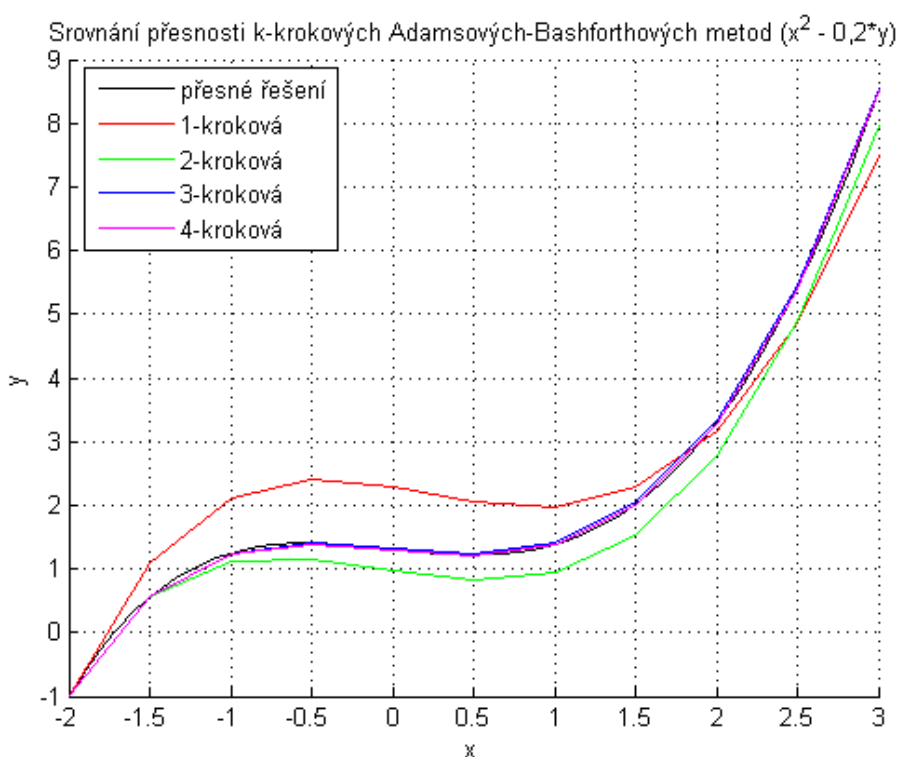
$i$	$x_i$	$y_i$
0	-2	-1
1	-1,5	0,5535
2	-1	1,2448
3	-0,5	1,3941

Následně se provede výpočet jednotlivých  $k$ -krokových metod.

**Tabulka 2.5** Výpočet pomocí Adams-Bashforthových  $k$ -krokových metod

i	$x_i$	Adams-Bashforthova metoda			
		1-kroková	2-kroková	3-kroková	4-kroková
0	-2	-1.0000	-1.0000	-1.0000	-1.0000
1	-1,5	1.1000	0.5535	0.5535	0.5535
2	-1	2.1150	1.1080	1.2448	1.2448
3	-0,5	2.4035	1.1570	1.4133	1.3941
4	0	2.2882	0.9763	1.3270	1.2994
5	0,5	2.0593	0.8252	1.2509	1.2149
6	1	1.9784	0.9377	1.4209	1.3793
7	1,5	2.2806	1.5258	2.0517	2.0055
8	2	3.1775	2.7814	3.3375	3.2881
9	2,5	4.8598	4.8779	5.4538	5.4022
10	3	7.4988	7.9728	8.5597	8.5067

Výsledné hodnoty se vykreslí pomocí grafu.



**Graf 2.2** Grafické řešení průběhu diferenciální rovnice pro jednotlivé  $k$ -krokové metody

Algoritmy výpočtů diferenciální rovnice jsou obsaženy v souborech AdamsBashf1krok.m, AdamsBashf2krok.m, AdamsBashf3krok.m a AdamsBashf4krok.m.

### 3 Identifikace a modelování systémů

V kapitole Identifikace a modelování systémů se zabývám rozdělením identifikace na analytickou a experimentální a zhodnocením jejich rozdílných přístupů k tvorbě modelů. Následně vysvětlím základní pojmy týkající se identifikace a modelování systémů a uvedu způsoby popisu systémů. Na závěr kapitoly uvedu na příkladu postup při převodu vnějšího popisu systému na vnitřní popis systému.

**Identifikace** představuje proces tvorby matematického modelu na základě reálného objektu. Cílem identifikace je nalezení modelu reálného objektu, který by co nejpřesněji popisoval chování reálného objektu.

Identifikaci systému rozdělujeme na:

- analytickou,
- experimentální.

Metodou **analytické identifikace** se sestavuje matematický model na základě matematicko-fyzikální analýzy zkoumaného objektu. Při analýze objektu se vychází z technologických, konstrukčních, ale také provozních dat zkoumaného objektu. Na základě znalostí fyzikálních, chemických nebo biologických dílčích procesů, které matematicky popisují jevy probíhající ve zkoumaném objektu, se získávají vztahy mezi sledovanými veličinami. Takto získané vztahy jsou nejčastěji popsány diferenciální nebo diferenční rovnicí. Aplikací těchto znalostí se získává matematický model zkoumaného objektu, který popisuje vnitřní stavové veličiny modelovaného procesu a jejich vzájemné vazby.

Nutno podotknout, že sledované a popisované dílčí procesy probíhající v modelovaném objektu jsou popsány pouze přibližně, jedná se o aproximaci reálného objektu. To znamená, že není možné dosáhnout úplné shody mezi chováním modelovaného objektu a jeho modelu. [DOSTÁL, 2006]

Na základě účelu použití modelu se zvažuje potřebná hloubka, s jakou jsou jevy sledovány, a také struktura popisovaného objektu. S narůstající hloubkou analýzy jevů se zpřesňuje chování matematického modelu, ale zároveň se zvyšuje složitost modelu a jeho

nákladnost. Pracnější je také odvození dílčích procesů i následné používání modelu. Z těchto důvodů je důležité zvážit a určit hloubku, do jaké je potřeba objekt analyzovat, aby byl dostatečně přesný, ale aby zároveň nebyl příliš složitý a nákladný.

Výhoda analytické identifikace spočívá v tom, že umožňuje simulovat vlastnosti objektu před jeho zhotovením. Výsledky simulace poté mohou ovlivnit výslednou realizaci zhotovení skutečného objektu.

Metoda **experimentální identifikace** oproti analytické identifikaci pracuje s existujícím objektem. Experimentální identifikace je založena na práci se skutečnými, naměřenými vstupními a výstupními hodnotami skutečného objektu, na jejich následném zpracování a vyhodnocení. Výsledkem těchto měření je experimentální model skutečného objektu. Získané parametry modelu charakterizují pouze vztahy mezi vstupem a výstupem, z tohoto důvodu je výsledný model modelem chování, nikoli struktury, jako v případě modelu získaného metodou analytické identifikace.

### 3.1 Základní pojmy

**Modelovaný systém** představuje zkoumaný reálný systém (reálný objekt, stroj, atd.), kterému byl podle daných kritérií přiřazen fyzický nebo abstraktní (virtuální) model.

**Fyzický model** je takový model, kterému je na základě geometrické podobnosti přiřazen model reálného objektu.

**Matematický model** je model abstraktní (virtuální), který umožňuje zkoumat jevy probíhající na reálném objektu pomocí matematického popisu jejich průběhů. Matematický model je většinou vyjádřen rovnicemi, např. diferenciálními, diferenčními, algebraickými.

**Simulací** nazýváme experiment na matematickém modelu reálného objektu.

**Simulační model** představuje odvozený matematický model, vytvořený pomocí programového simulačního prostředí. Pomocí simulačního modelu lze sledovat chování systému a zjišťovat hodnoty potřebných fyzikálních veličin.

**Modelování** je proces tvorby simulačního modelu.

**Systémem** rozumíme obecně soubor prvků, mezi nimiž existují vzájemné vazby a jako celek má určité vztahy ke svému okolí.

Na základě definice systému existují **vazby**:

- vnitřní – vazby mezi prvky systému,
- vnější – vazby mezi prvky systému a jeho okolím.

Poté části systému jsou nazývány **prvky** a stejně jako vazby se dělí na:

- vnitřní – prvek nemá spojení s okolním systémem,
- vnější – prvek má spojení s okolním systémem.

Každý prvek systému může být systémem nižšího řádu (podsystem), stejně tak jako každý systém může být prvkem systému vyššího řádu (nadsystem). Tato vlastnost souvisí s rozlišovací úrovní.

**Rozlišovací úroveň** představuje míru podrobnosti, s jakou je systém popsán.

**Okolí systému** je množina prvků, které nejsou přímo prvky daného systému, ale mají se systémem určité vazby.

### 3.2 Analytická identifikace systémů - popis systémů

Dynamicke vlastnosti systému je možné popsat několika různými způsoby, které lze rozdělit do dvou skupin na

- vnější popis systému,
- vnitřní popis systému.

Vnější popis dynamického systému vyjadřuje závislost mezi výstupní veličinou  $y(t)$  a vstupní veličinou  $u(t)$  bez použití vnitřních (stavových) veličin  $x(t)$ .

Vnitřní popis vyjadřuje dynamicke vlastnosti systému v časové oblasti. Vnitřní popis vede na stavový model systému, který je popsán stavovou rovnicí (3.3) a výstupní rovnicí (3.4).

#### Diferenciální rovnice

Diferenciální rovnice představuje matematický popis systému v časové oblasti. Řád systému je dán řádem diferenciální rovnice. Popis vlastností systému je dán obecným tvarem lineární diferenciální rovnice s konstantními koeficienty

$$\begin{aligned}
 a_n y^n(t) + a_{n-1} y^{n-1}(t) + \dots + a_1 y'(t) + a_0 y(t) = \\
 = b_m u^m(t) + b_{m-1} u^{m-1}(t) + \dots + b_1 u'(t) + b_0 u(t),
 \end{aligned}
 \tag{3.1}$$

kde  $a_n, b_m$  jsou konstantní koeficienty,  $u(t)$  je vstupní veličina a  $y(t)$  je výstupní veličina.

K řešení diferenciálních rovnic je nutné znát počáteční podmínky rovnice  $y(0), y'(0), \dots, y^{n-1}(0)$  a  $u(0), u'(0), \dots, u^{m-1}(0)$ .

Podmínky fyzikální realizovatelnosti:

- silná  $n > m$ ,
- slabá  $n = m$ ,
- nerealizovatelný  $n < m$ .

Z rovnic fyzikální realizovatelnosti systému vyplývá, že nejvyšší stupeň derivace výstupní veličiny musí být větší nebo roven stupni derivace vstupní veličiny.

### **Obrazový přenos**

Obrazový přenos  $G(s)$  je definován jako poměr Laplaceova obrazu výstupní veličiny  $Y(s)$  k Laplaceovu obrazu vstupní veličiny  $U(s)$  při nulových počátečních podmínkách

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}.
 \tag{3.2}$$

Podmínky fyzikální realizovatelnosti:

- silná  $n > m$ ,
- slabá  $n = m$ ,
- nerealizovatelné  $n < m$ .

Z rovnic fyzikální realizovatelnosti systému vyplývá, že nejvyšší stupeň derivace výstupní veličiny musí být větší nebo roven stupni derivace vstupní veličiny. Řádem  $n$  diferenciální rovnice je určen řád systému.

### **Stavový popis**

Lineární spojitý dynamický systém je popsán současně stavovou rovnicí  $\dot{\mathbf{x}}(t)$  a výstupní rovnicí  $y(t)$  ve tvaru

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \quad (3.3)$$

$$y(t) = \mathbf{c}^T \mathbf{x}(t) + du(t) \quad (3.4)$$

kde

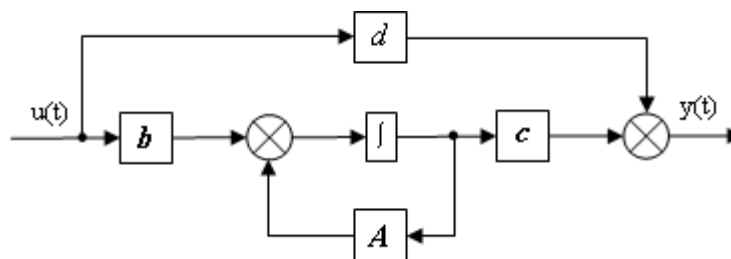
$\mathbf{A}$  - je stavová matice systému, rozměr  $(n, n)$

$\mathbf{b}$  - je stavový vektor řízení, rozměr  $(n, 1)$

$\mathbf{c}$  - je výstupní vektor systému, rozměr  $(1, n)$

$d$  - je konstanta převodu

Podle rovnic stavového popisu (3.3 a 3.4) lze sestavit obecné blokové schéma spojitého lineárního dynamického systému (obrázek 3.1).



Obrázek 3.1 Obecné blokové schéma spojitého lineárního dynamického systému

Ze stavových rovnic a ze schématu stavového popisu (obrázek 3.1) vyplývá, že k realizaci lineárního spojitého systému je potřeba bloky typu:

- integrátor (počet je roven řádu systému  $n$ ),
- zesilovač,
- sumační člen.

Podmínky fyzikální realizovatelnosti pro stavový model má následující tvar:

- silná  $d = 0$ ,
- slabá  $d \neq 0$  ( $n = m$ ).

Systém, který splňuje silnou podmínku fyzikální realizovatelnosti je takový, u kterého výstup závisí pouze na stavových veličinách a rovnice výstupu (3.4) neobsahuje vstupní veličiny. Pro tyto systémy je konstanta převodu  $d$  nulová.

Je-li dynamický systém popsáný vnějším popisem (diferenciální rovnicí nebo obrazovým přenosem), pak se provádí jeho převod na vnitřní stavový popis systému jednou z následujících metod:

- metoda postupné integrace,
- metoda snižování řádu derivace.

[NOSKIEVIČ, 1992]

Postup převodu dynamického systému popsáného vnějším popisem na vnitřní stavový popis systému pomocí výše uvedeným metod je uveden v následujících kapitolách.

Po konzultaci s vedoucí bakalářské práce jsem jako vzorový příklad sestavení modelu dynamického systému pomocí metod postupné integrace a snižování řádu derivace zvolil lineární diferenciální rovnici třetího řádu s nenulovými počátečními podmínkami.

Na základě vypočtených stavových proměnných a jejich počátečních podmínek jsem sestavil bloková schémata a podle blokových schémat jsem následně vytvořil simulační model v pracovním prostředí Simulink. Výsledná schémata jsou uvedena v kapitole Simulink (kap. 4.2).

Postup tvorby blokových schémat simulačního modelu je názorně prezentován pomocí interaktivních prezentací vytvořených v programu Adobe Captivate. Interaktivní prezentace jsou součástí elektronických výukových materiálů.

### 3.3 Metoda postupné integrace

Metodu postupné integrace je vhodné použít pro rovnice s derivacemi vstupu  $x$  na pravé straně. Nejvyšší řád derivace vstupu  $x$  musí být menší než nejvyšší řád derivace  $y$ .

Postupnou integrací diferenciální rovnice a stanovením stavových proměnných je nejprve provedeno sestavení soustavy stavových proměnných, z nichž je navrženo blokové schéma.

#### **Příklad 3.1**

Je dán model lineárního dynamického systému zadaného diferenciální rovnicí ve tvaru  $0,5y'''(t) + 4y''(t) + 5y'(t) + y(t) = 3u(t)$  s obecným vstupním signálem. Počáteční



podmínky jsou nenulové:  $y''(0) = 2$ ,  $y'(0) = 1$  a  $y(0) = -3$ . Pomocí metody postupné integrace určete vnitřní stavový popis daného systému.

### Řešení:

Je dána lineární diferenciální rovnice

$$0,5y'''(t) + 4y''(t) + 5y'(t) + y(t) = 3u(t)$$

s počátečními podmínkami

$$y''(0) = 2, \quad y'(0) = 1, \quad y(0) = -3.$$

- Nejprve osamostatníme nejvyšší řád derivace, členy se stejným řádem je vhodné psát vedle sebe

$$0,5y'''(t) + 4y''(t) + 5y'(t) + y(t) = 3u(t)$$

$$0,5y'''(t) = -4y''(t) - 5y'(t) - y(t) + 3u(t)$$

$$y'''(t) = -8y''(t) - 10y'(t) - 2y(t) + 6u(t)$$

- Provedeme integraci obou stran rovnice, tím se sníží řád derivace

$$y''(t) - y''(0) = -8y'(t) + 8y'(0) - 10y(t) + 10y(0) + \int_0^t [-2y(\tau) + 6u(\tau)] d\tau$$

- Upravíme proměnné a zavedením substituce integrálu stanovíme první stavovou proměnnou  $x_1$

$$y''(t) = -8y'(t) - 10y(t) + y''(0) + 8y'(0) + 10y(0) + \int_0^t [-2y(\tau) + 6u(\tau)] d\tau$$

$$x_1 = \int_0^t [-2y(\tau) + 6u(\tau)] d\tau + x_{10}$$

- Stanovíme počáteční podmínku  $x_0$  tak, aby kompenzovala hodnoty počátečních podmínek  $y$  a  $u$

$$x_{10} = y''(0) + 8y'(0) + 10y(0)$$

- Po dosazení získáme rovnici

$$y''(t) = -8y'(t) - 10y(t) + x_1$$

- Získali jsme diferenciální rovnici druhého řádu a opakujeme předchozí postup (integrace, úprava, substituce stavové proměnné  $x_2$ )

$$y'(t) - y'(0) = -8y(t) + 8y(0) + \int_0^t [-10y(\tau) + x_1] d\tau$$

$$y'(t) = -8y(t) + y'(0) + 8y(0) + \int_0^t [-10y(\tau) + x_1] d\tau$$

$$x_2 = \int_0^t [-10y(\tau) + x_1] d\tau + x_{20}$$

$$x_{20} = y'(0) + 8y(0)$$

- Po dosazení získáme rovnici

$$y'(t) = -8y(t) + x_2$$

- Získali jsme diferenciální rovnici prvního řádu a opakujeme předchozí postup (integrace, úprava, substituce stavové proměnné  $x_3$ )

$$y(t) - y(0) = \int_0^t [-8y(\tau) + x_2] d\tau$$

$$y(t) = y(0) + \int_0^t [-8y(\tau) + x_2] d\tau$$

$$x_3 = \int_0^t [-8y(\tau) + x_2] d\tau + x_{30}$$

$$x_{30} = y(0)$$

- Po dosazení získáme rovnici

$$y(t) = x_3$$

Postupnou integrací jsme získali všechny stavové proměnné, které jsou určující pro sestavení blokového schématu při řešení diferenciální rovnice metodou postupné integrace

$$x_1 = \int_0^t [-2y(\tau) + 6u(\tau)] d\tau + x_{10}$$

$$x_2 = \int_0^t [-10y(\tau) + x_1] d\tau + x_{20}$$

$$x_3 = \int_0^t [-8y(\tau) + x_2] d\tau + x_{30}$$

Na základě získaných stavových proměnných sestavíme blokové schéma a matematický model (kap. 4.2).

### 3.4 Metoda snižování řádu derivace

Druhou metodou pro tvorbu modelu dynamického systému je metoda postupného snižování řádu derivace. Na levé straně rovnice se osamostatní nejvyšší derivace výstupu. Sériovým zapojením příslušného počtu integrátorů se postupně odstraní derivace proměnné, nižší řády se využijí k sestavení dílčích částí pravé strany rovnice.

Všeobecnou podmínkou řešitelnosti diferenciální rovnice je, že pravá strana rovnice nesmí obsahovat vyšší derivaci vstupního signálu než strana levá.

Aplikace metody snižování řádu derivace je ukázána na stejném příkladu lineární diferenciální rovnice, včetně stejných nenulových podmínek jako v případě metody postupné integrace. Pak lze navzájem porovnat jednotlivé modely sestavení oběma metodami a porovnat rozdíly ve struktuře blokového schématu.

#### **Příklad 3.2**

Je dán model lineárního dynamického systému zadaného diferenciální rovnicí ve tvaru  $0,5y'''(t) + 4y''(t) + 5y'(t) + y(t) = 3u(t)$  s obecným vstupním signálem. Počáteční podmínky jsou nenulové:  $y''(0) = 2$ ,  $y'(0) = 1$  a  $y(0) = -3$ . Pomocí metody snižování řádu derivace určete vnitřní stavový popis daného systému.

#### **Řešení:**

Je dána lineární diferenciální rovnice

$$0,5y'''(t) + 4y''(t) + 5y'(t) + y(t) = 3u(t)$$

s počátečními podmínkami

$$y''(0) = 2, \quad y'(0) = 1, \quad y(0) = -3.$$

- Počet stavových proměnných odpovídá řádu nejvyšší derivace výstupní veličiny řešené diferenciální rovnice. V tomto případě stavový popis využívá stavové proměnné  $x_1$ ,  $x_2$  a  $x_3$ .
- Stavová proměnná  $x_1$  je rovna výstupní veličině

$$x_1 = y$$

- Každá následující stavová proměnná je rovna derivaci předcházející stavové proměnné (výstupní veličině)

$$x_2 = y'$$

$$x_3 = y''$$

- Nejvyšší derivace je rovna derivaci nejvyšší stavové proměnné

$$\dot{x}_3 = y'''$$

- Z původní diferenciální rovnice vyjádříme  $y'''$

$$y'''(t) = -8y''(t) - 10y'(t) - 2y(t) + 6u(t)$$

- Za jednotlivé derivace výstupní veličiny dosadíme do rovnice derivace stavových proměnných

$$x_1 = y$$

$$x_2 = \dot{x}_1$$

$$x_3 = \dot{x}_2$$

$$\dot{x}_3 = -8x_3 - 10x_2 - 2x_1 + 6u(t)$$

- Počáteční podmínky stavových proměnných jsou přímo rovny počátečním podmínkám výstupu  $y(0)$ ,  $y'(0)$  a  $y''(0)$ , jak vyplývá ze vztahů stavových proměnných

$$x_{10} = y(0) = -3$$

$$x_{20} = y'(0) = 1$$

$$x_{30} = y''(0) = 2$$

Blokové schéma a matematický model řešení diferenciální rovnice pomocí metody snižování řádu derivace jsou uvedeny v kapitole 4.2.

## 4 Popis systému MATLAB<sup>®</sup> a nadstavby Simulink<sup>®</sup>

V této kapitole se zaměřuji na popis programu MATLAB a jeho nadstavby Simulink, který jsem použil při tvorbě simulačních modelů vzorového příkladu.

### 4.1 MATLAB

MATLAB je integrované prostředí založené na maticovém kalkulu zahrnující nástroje pro vědeckotechnické výpočty, analýzu dat a jejich vizualizaci, modelování a simulaci dějů, měření a zpracování signálů. Název programu vznikl zkrácením anglických slov MATrix LABoratory.

MATLAB poskytuje svým uživatelům k využití silné grafické a výpočetní nástroje, rozsáhlé sady úzce zaměřených knihoven funkcí a výkonný programovací jazyk čtvrté generace. Knihovny funkcí jsou vzhledem ke svému rozsahu využitelné prakticky ve všech oborech lidské činnosti.

Původně MATLAB vznikl jako nadstavba maticových knihoven LINPACK a EISPACK a pro usnadnění práce s nimi. Nově definované funkce mohou být interpretovány buď přímo z textové podoby souborů, nazývané jako m-soubory (m-files), nebo z předzpracované podoby p-soubory (p-files). Systém také umožňuje přidávat zkompilevané moduly v podobě souboru mex-files do strojového kódu procesoru. Vlastní programovací jazyk MATLABu vychází z programovacího jazyka Fortran.

Grafický výstup má v prostředí MATLAB velice silnou podporu jak pro vizualizaci dat, tak pro tvorbu uživatelského prostředí. Mezi základní funkce patří vizualizace dat prostřednictvím 2D (dvourozměrných) a 3D (třírozměrných) grafů s velkým množstvím volitelných parametrů. Parametry lze jednoduše měnit nejen při samotném vytváření grafu, ale i po jeho vytvoření.

Důležitou částí instalace MATLABu jsou knihovny funkcí (adresáře s m a mex soubory), které jsou nazývány toolboxy. Toolboxy obsahují uceleným způsobem, včetně dokumentace a příkladů, zpracovaný určitý obor numerické matematiky, analytické matematiky, statistiky, systémového přístupu k regulacím a další obory, ve kterých nachází MATLAB uplatnění.

## 4.2 Simulink

Simulink představuje jednu z několika nadstaveb systému MATLAB. Simulink je nadstavba MATLABu pro simulaci a modelování dynamických systémů, který využívá algoritmy MATLABu pro numerické řešení nelineárních diferenciálních rovnic s grafickým zadáváním řešené soustavy. Simulink umožňuje snadno a rychle vytvářet modely dynamických soustav ve formě blokových schémat a rovnic. S jeho pomocí lze určit časové průběhy výstupních veličin, stejně jako veličin vnitřních v závislosti na časovém průběhu vstupních veličin a počátečním stavu.

Simulink není součástí MATLABu od počátku, vznikl až později a jeho první verze byla součástí MATLABu až pro verzi MATLAB 4. Uživatelské prostředí Simulinku je odlišné oproti MATLABu, který je založen na práci s příkazovým řádkem.

Princip práce v Simulinku je založen na blocích, které reprezentují základní dynamické systémy. Propojení příslušných signálových vstupů a výstupů bloků z příslušné knihovny vznikají modely systémů. Pro vstupy modelu se nejčastěji používají signály z bloků knihoven, které generují základní typy signálů a které lze libovolně poskládat. Simulink umožňuje vytvářet tzv. subsystémy, které vzniknou sloučením libovolného počtu bloků do skupiny. Takto vzniklému subsystému lze definovat externí vstupy a výstupy. Dále se s tímto subsystémem pracuje jako se základním blokem. Výsledek simulace, jako výstup je ve formě časového průběhu řešení a nejčastěji se zobrazuje graficky pomocí standardních bloků, jako je osciloskop nebo 2D graf.

Prostředí pro tvorbu a simulaci modelů se spouští z příkazové řádky systému Matlab příkazem `simulink`. Po spuštění se vytvoří okno pro vytváření nového modelu a okno Simulink Library Browser obsahující základní nabídku knihoven zdrojů signálů, základních spojitých, diskretních a nelineárních bloků a bloků pro zobrazování a ukládání signálů.

**Příklad 4.1**

K matematicky získanému výsledku řešení diferenciální rovnice (Příklad 3.1)

$$x_1 = \int_0^t [-2y(\tau) + 6u(\tau)] d\tau + x_{10}$$

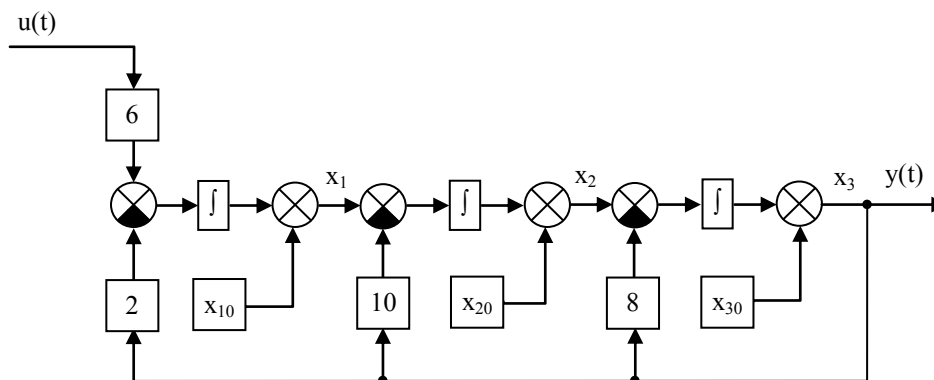
$$x_2 = \int_0^t [-10y(\tau) + x_1] d\tau + x_{20}$$

$$x_3 = \int_0^t [-8y(\tau) + x_2] d\tau + x_{30}$$

pomocí metody postupné integrace (kap. 3.3) sestavte blokové schéma a vytvořte simulační model v prostředí Simulink.

**Řešení:**

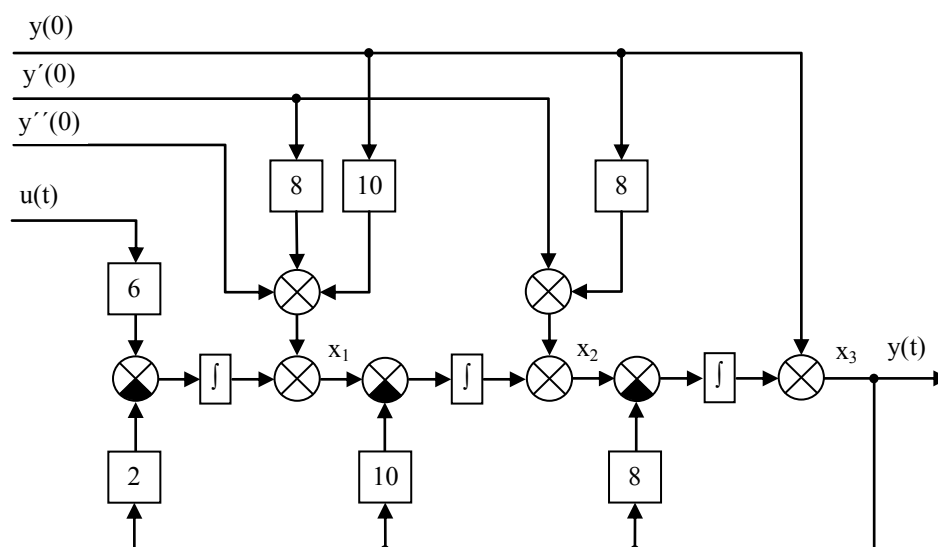
Na základě získaných rovnic stavových proměnných sestavíme blokové schéma, které je zobrazeno na obrázku 4.1.



**Obrázek 4.1 Metoda postupné integrace - blokové schéma s konstantními počátečními podmínkami**

Můžeme sestavit také rozšířené blokové schéma o výpočet počátečních podmínek  $x_{10}$ ,  $x_{20}$  a  $x_{30}$ . Takové blokové schéma je na obrázku 4.2.



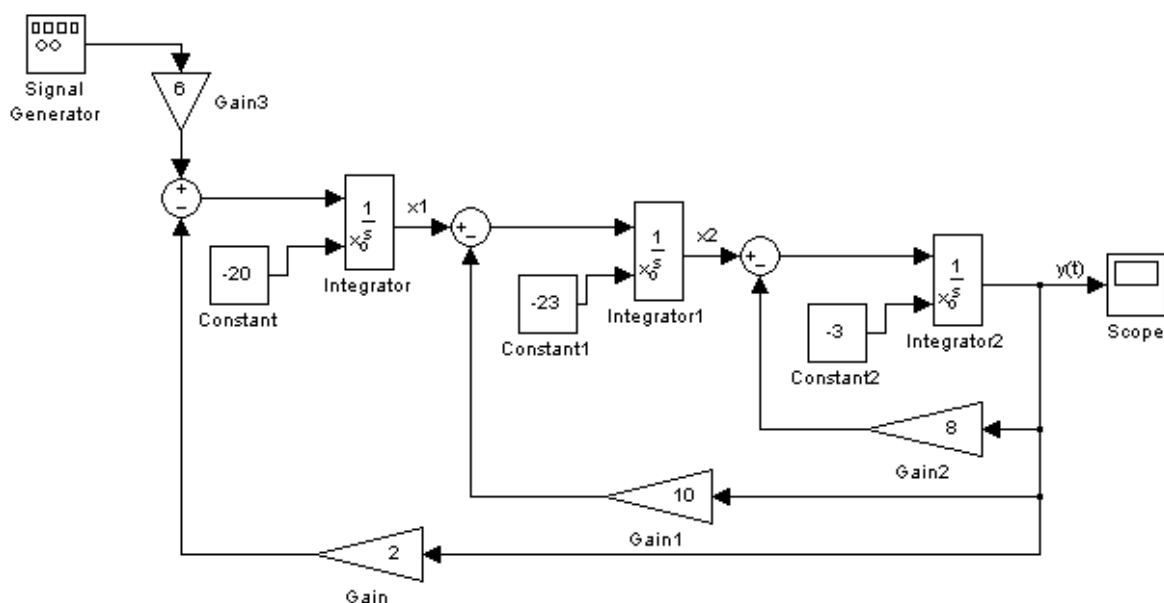


**Obrázek 4.2 Metoda postupné integrace - blokové schéma rozšířené o výpočet počátečních podmínek**

Při sestavování blokových schémat je vhodné dodržet tato pravidla:

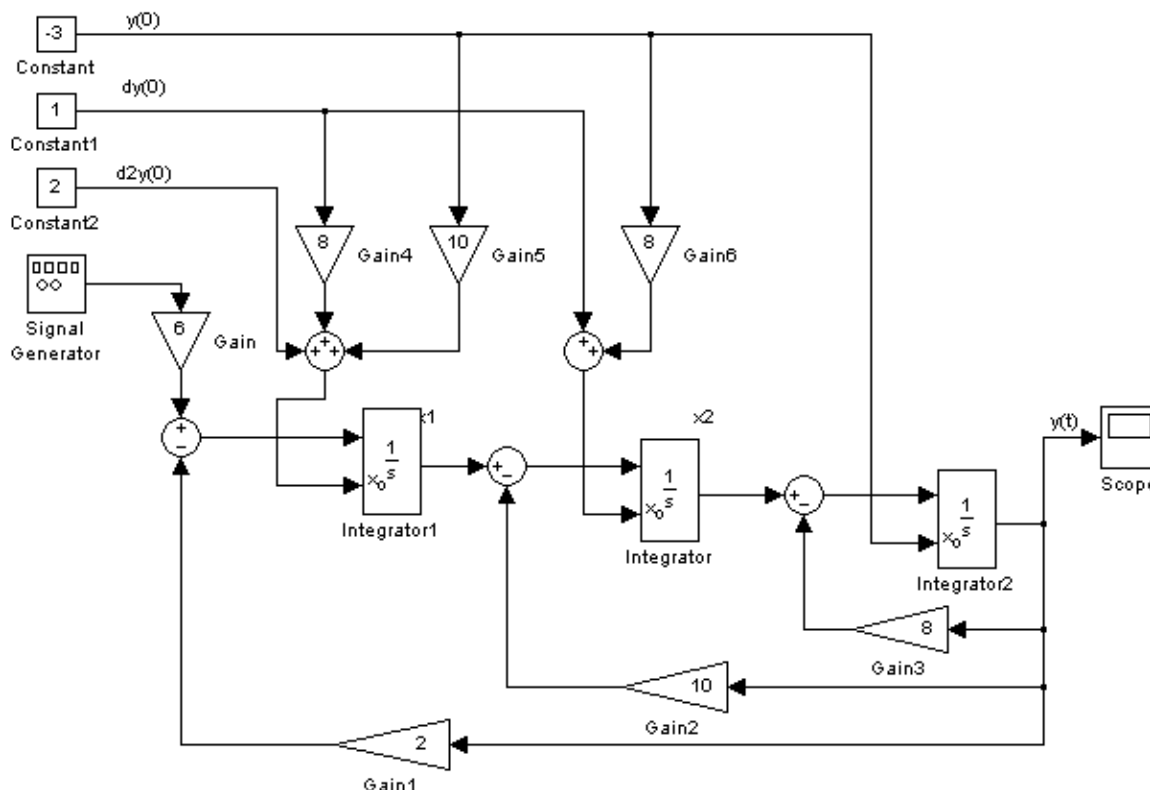
- počáteční podmínky  $y''(0)$ ,  $y'(0)$  a  $y(0)$  vedou zleva stejně jako vstup  $u(t)$ ,
- vpravo směřuje pouze výstup  $y(t)$ .

Podle blokového schématu na obrázku 4.1 sestavíme simulační model v programu Simulink, (obrázek 4.3).



**Obrázek 4.3 Metoda postupné integrace - simulační model s konstantními počátečními podmínkami v programu Simulink**

Pro blokové schéma rozšířené o výpočet počátečních podmínek  $x_{10}$ ,  $x_{20}$  a  $x_{30}$  (obrázek 4.2) je simulační model znázorněn schématem na obrázku 4.4.



Obrázek 4.4 Metoda postupné integrace - simulační model v programu Simulink rozšířený o výpočet počátečních podmínek

Na základě matematického řešení diferenciální rovnice pomocí metody postupné integrace jsem pro dané řešení sestavil blokové schéma a matematický model v programu Simulink. Postup sestavení matematického modelu je popsán pomocí interaktivní prezentace vytvořené v programu Adobe Captivate. Interaktivní prezentace je součástí elektronických výukových materiálů.

### Příklad 4.2

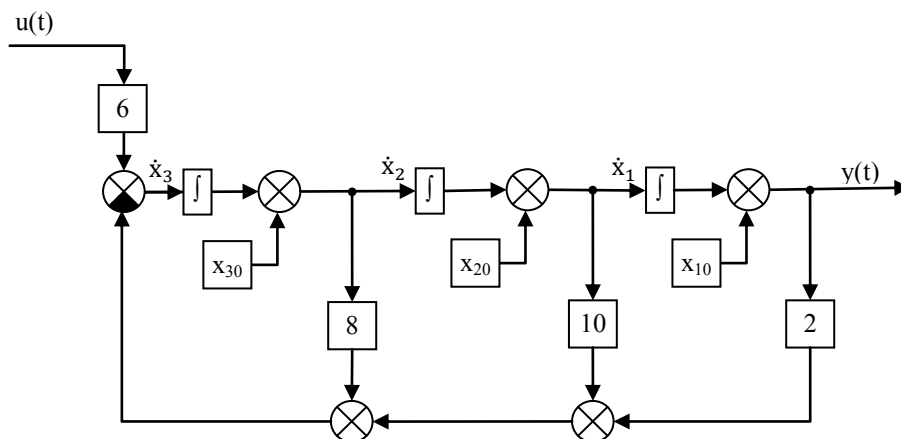
K matematicky získanému výsledku řešení diferenciální rovnice (Příklad 3.2)

$$\dot{x}_3 = -8x_3 - 10x_2 - 2x_1 + 6u(t)$$

pomocí metody snižování řádu derivace (kap. 3.4) sestavte blokové schéma a vytvořte simulační model v programu Simulink.

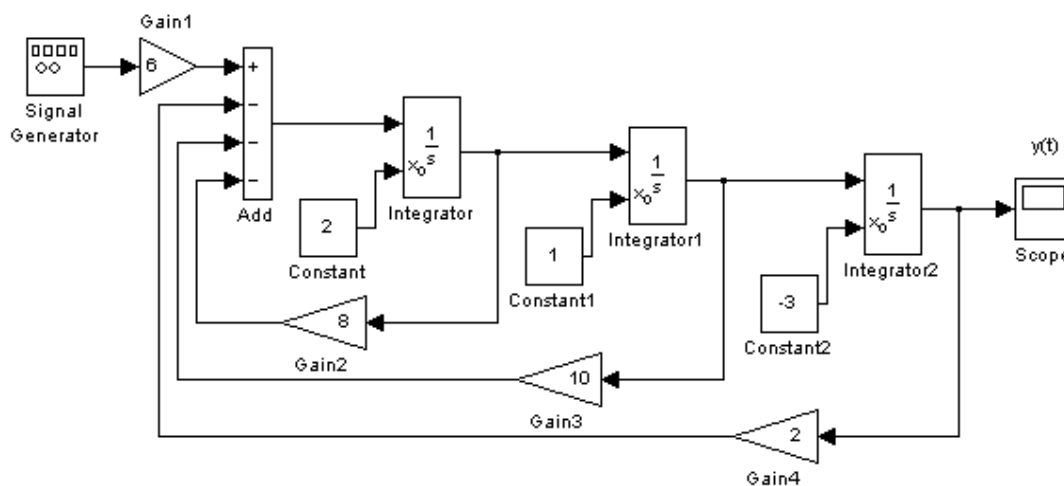
**Řešení:**

Na základě získaných rovnic stavových proměnných sestavíme blokové schéma, které je zobrazeno na obrázku 4.5.



**Obrázek 4.5 Metoda snižování řádu derivace - blokové schéma**

Podle blokového schématu na obrázku 4.5 sestavíme simulační model v programu Simulink, tento simulační model je na obrázku 4.6.



**Obrázek 4.6 Metoda snižování řádu derivace - simulační model s v programu Simulink**

Blokové schéma a matematický model jsem sestavil na základě matematického řešení diferenciální rovnice pomocí metody snižování řádu derivace. Postup sestavení matematického modelu je popsán pomocí interaktivní prezentace vytvořené v programu Adobe Captivate. Interaktivní prezentace je součástí elektronických výukových materiálů.

## 5 Seznámení s programem Adobe Captivate

Adobe Captivate je profesionální softwarový program na tvorbu multimediálního výukového obsahu v oblasti e-learning, elektronických manuálů a interaktivních prezentací kombinujících statické snímky, video a zvukové záznamy. Hlavní součástí Adobe Captivate tvoří sofistikovaný modul pro pořizování obrazových záznamů včetně akcí prováděných myší či klávesnicí.

Práce s programem je založena na snímání obrazovky. Snímání lze nastavit buď na celou obrazovku, na okno spuštěné aplikace nebo si lze předem nadefinovat velikost oblasti nahrávání a toto pomyslné okno umístit podle potřeby kdekoli na obrazovce.

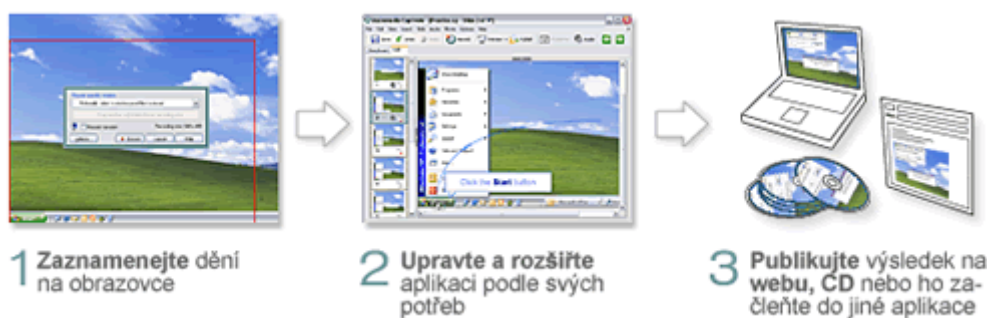
Veškerá automaticky zaznamenaná činnost prováděná na obrazovce počítače je následně programem převáděna do instruktážní prezentace formou animace nebo do interaktivní simulace práce s programem.

Vytvořené nahrávky je možné následně v editačním režimu upravovat a doplňovat o další prvky, kterými mohou být:

- Textové popisy, grafika, mluvený komentář nebo hudba.
- Flash animace a video.
- Interaktivní prvky.
- Znalostní testy a jejich vyhodnocení.

V programu Adobe Captivate je implicitně nastavený výstupní formát ve formě SWF souboru. Ve skutečnosti nabízí Adobe Captivate celou řadu možností, jak výslednou práci uložit, exportovat nebo prezentovat jako jsou PDF soubory, export na YouTube server či výsledné video soubory AVI, MPEG a podobně.

Popsaný postup práce s Adobe Captivate je znázorněn graficky na obrázku 5.1.

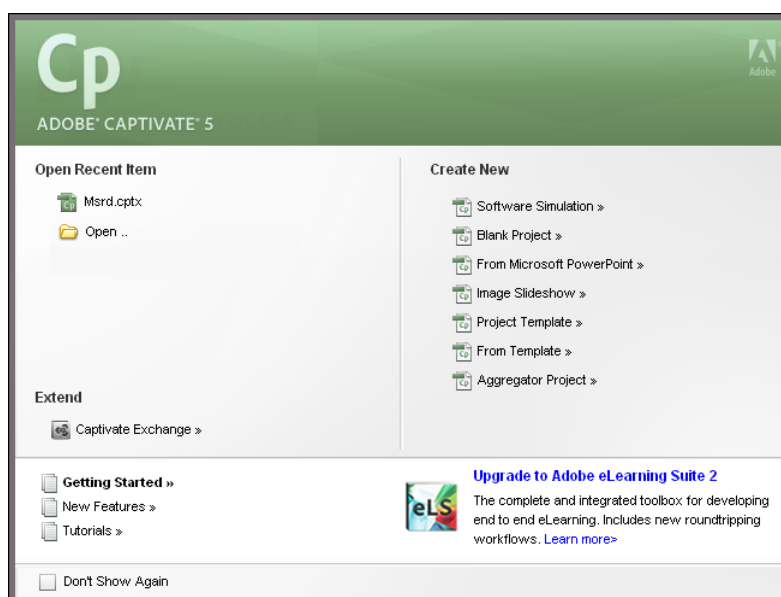


Obrázek 5.1 Schématický postup práce s Adobe Captivate (převzato z [www.digitalmedia.cz](http://www.digitalmedia.cz))

## 5.1 Tvorba nového projektu

Nejběžnější způsob vytvoření nového projektu je otevřením menu *File > Record New Project*. Adobe Captivate nabídne úvodní obrazovku, která je rozdělena na tři oblasti (obrázek 5.2).

- „Open Recent File“ – otevření existujícího projektu
- „Create New“ – vytvoření nového projektu
- Oblast nápovědy – odkazy na manuály



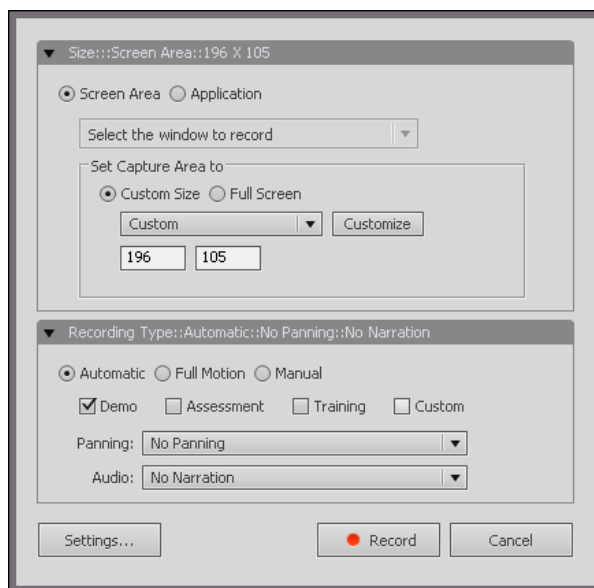
Obrázek 5.2 Úvodní obrazovka programu Adobe Captivate

Nabídka „Create New“ na úvodní obrazovce nabízí několik způsobů vytvoření nového projektu

- „Software Simulation“ – snímání obrazovky.
- „Blank Project“ – prázdný projekt, umožňuje volnost při vytváření prezentace.

- „From Microsoft PowerPoint“ – import prezentace z Microsoft PowerPoint.
- „Image Slideshow“ – vytváření prezentace z obrázků (fotografií).
- „Project Template“ – výroba šablony projektu.
- „From Template“ – vytváření projektu podle šablony.

Z důvodu potřeby využití programu Adobe Captivate pro tvorbu interaktivních prezentací je dále uveden podrobněji pouze popis režimu „Software Simulation“. Po výběru a spuštění nového projektu zobrazí Adobe Captivate okno (obrázek 5.3), ve kterém lze nastavit možnosti snímání.



Obrázek 5.3 Volba plochy snímání a způsobu nahrávání

Možnosti výběru jsou následující

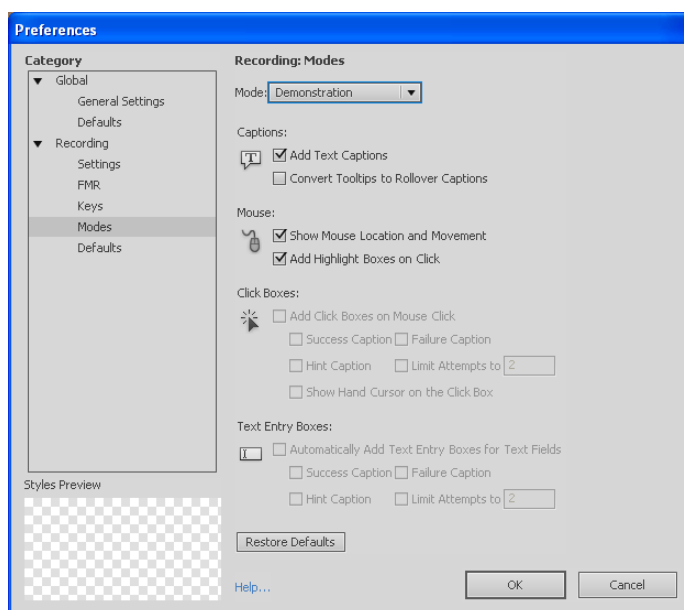
- Screen Area – libovolné nastavení velikosti snímané plochy.
- Application – automatické nastavení velikosti snímané plochy podle zvoleného programu.

V dolní části okna se nachází možnosti způsobu nahrávání

- Demonstration mode
  - snímání postupu činnosti s pasivním výstupem,
  - pozorovateli je umožněno pouze sledovat zaznamenané činnosti bez možnosti zásahu při sledování.
- Training mode
  - umožňuje aktivní účast pozorovatele,

- posun v prezentaci je umožněn po provedení nadefinovaných akcí a činností.
- Assessment mode
  - je určen k ověřování znalostí pozorovatele během sledování prezentace.
- Custom mode
  - používá se při vytváření projektu, kde se kombinují ostatní módy.

Rozšířené možnosti nastavení je možno upravit v okně aktivovaném tlačítkem *Settings* (obrázek 5.4).



Obrázek 5.4 Rozšířené možnosti nastavení

Při tvorbě interaktivních prezentací jsem volil Demonstration mode bez možnosti aktivních zásahů pozorovatele během sledování prezentace.

## 5.2 Editace projektu

Snímání činnosti se ukončí přepnutím do programu Adobe Captivate, nebo lze snímání ukončit klávesou **End**. Program se přepne do režimu zpracování projektu. Pracovní plocha programu je rozdělena do několika samostatných částí:

- Okno náhledu a úprav projektu – slouží k provádění úprav.
- Filmový pás – obsahuje soubor jednotlivých snímků projektu.
- Časová osa – umožňuje editaci a ladění interaktivních prvků snímku.

Při zpracování prezentací sestavení matematického modelu jsem postupoval podle předem sestaveného postupu:

- Spuštění toolboxu Simulink z programu MATLAB.
- Vytvoření nového projektu.
- Tvorba simulačního modelu s použitím:
  - integračního členu,
  - sumačního členu,
  - zesílení,
  - konstant,
  - zdroje signálu,
  - výstupního prvku matematického modelu.
- Uložení projektu.

### 5.3 Publikování projektu

Pro ověřování dílčího zpracování projektu lze použít možnost náhledu, který je k dispozici ve více variantách spuštění:

- Celý projekt v samostatném okně.
- Projekt lze spustit od určitého snímku do konce.
- Projekt lze spustit jen pro následujících 5 snímků.
- Celý projekt lze spustit v internetovém prohlížeči.

Adobe Captivate umožňuje publikování projektu do samostatně spustitelných prezentací různými způsoby. Publikování otevřeme v nabídce *File > Publish*. Pro publikování hotového projektu preferuji animaci typu Flash, kterou exportuji do HTML formátu. Vytvořená Flash animace je součástí elektronických výukových materiálů.



## 6 Návrh a tvorba elektronických výukových podkladů

Cílem bakalářské práce je vytvoření elektronické publikace jako podpory výuky k identifikaci a simulaci systémů ve formě HTML stránek. Kromě textové části je elektronická publikace doplněna také o zdrojové kódy vzorových příkladů programu MATLAB & Simulink. Postupy řešení jsou zaznamenány prostřednictvím programu Adobe Captivate a jsou také součástí elektronické publikace.

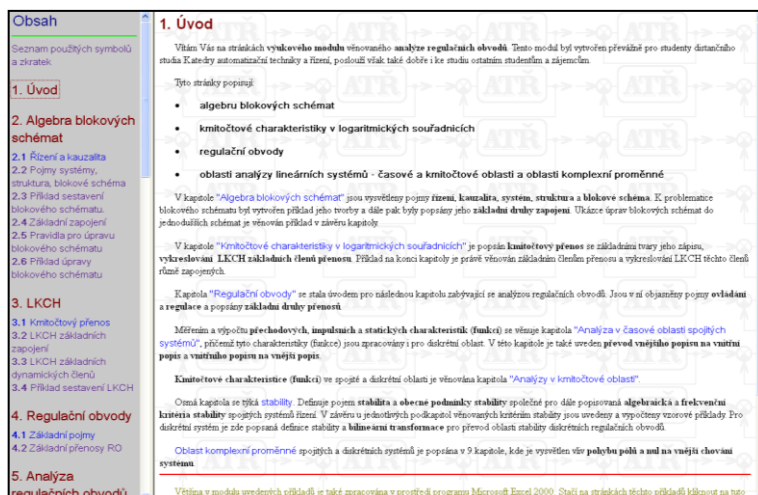
### 6.1 Elektronické publikace dostupné na stránkách katedry ATŘ

Před samotným návrhem elektronických podkladů k výukovému textu, jsem se v souladu s jedním z bodů v zadání bakalářské práce nejdříve seznámil s vybranými elektronickými publikacemi z oblasti automatického řízení. Elektronické publikace jsou dostupné na stránkách katedry automatizační techniky a řízení (ATŘ) na adrese [http://www.352.vsb.cz/uc\\_texty/Welcome.htm](http://www.352.vsb.cz/uc_texty/Welcome.htm).

Pro seznámení a následné zhodnocení již realizovaných výukových textů jsem si vybral celkem tři elektronické publikace.

#### Analýza lineárních regulačních obvodů

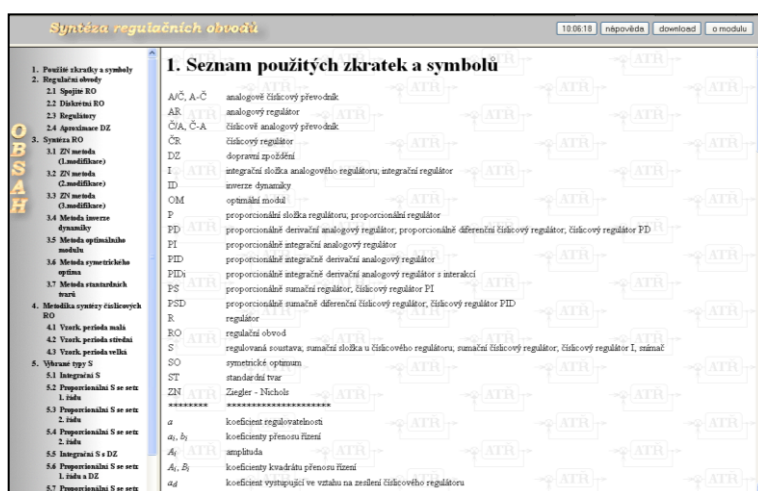
Elektronická publikace (obrázek 6.1) je založena na rámcovém rozvržení www stránky s dvěma sloupci. Jedná se o rozsáhlý učební text, jehož součástí je velký počet obrázků, které napomáhají studentům k lepšímu pochopení látky. Součástí řešených příkladů jsou na konci řešení umístěny odkazy na soubor se zpracovanými daty k danému problému. Jako nevhodné mi přijde řešení samotného obsahu, které je celý rozbalený s nevhodně zvolenou velikostí písma. Výsledkem toho je, že text není optimalizován pro standardní rozlišení monitorů a přetéká. Student je v tomto případě nucen neustále rolovat. Pozadí hlavní stránky, které je tvořeno logem ATŘ působí při práci s textem rušivě, kdy dochází místy až k splývání textu s pozadím.



Obrázek 6.1 Analýza lineárních regulačních obvodů – úvodní obrazovka

### Syntéza lineárních regulačních obvodů

Stejně jako předchozí, tak i tato elektronická publikace (obrázek 6.2) je založena na rámcovém rozvržení www stránky. U této publikace je ovšem rozvržení standardně rozděleno do tří částí: „Hlavička“ (rám ve formě horního řádku) s názvem učebního textu, „Menu“ (levý sloupec) s obsahem učebního textu, a samotný obsah učebního textu v rámu „Obsah“. Samotný učební text je velmi přehledný a opět doplněn řadou obrázků v podobě grafů. Na pozadí je opět použito logo ATŘ, v tomto případě nepůsobí až tak rušivě jako u předchozí elektronické publikace. Zřejmě je to dáno tím, že logo je menší a má mezi sebou větší mezery.



Obrázek 6.2 Syntéza lineárních regulačních obvodů – úvodní obrazovka

### Analýza a syntéza lineárních regulačních obvodů pomocí Delta transformace

Tato elektronická publikace (obrázek 6.3) mě velmi zaujala pro svůj jednoduchý a přehledný vzhled. Založena je na rámech v podobě dvou sloupců, pro menu a pro samotný učební text.



**Obrázek 6.3 Analýza a syntéza lineárních regulačních obvodů pomocí Delta transformace – úvodní obrazovka**

## 6.2 Tvorba v HTML editoru

V dnešní době je k dispozici spousta kvalitních programů pro vytváření www stránek a to i mezi programy, které se licenčně řadí mezi volně dostupné (freeware). Programy se dělí do dvou skupin na tzv. WYSIWYG editory a editory strukturní. Při tvorbě elektronické publikace byl použit software Microsoft Expression Web 4, který se řadí mezi WYSIWYG editory.

### WYSIWYG editor

Název editorů vychází z počátečních písmen anglických slov what you see is what you get, v českém překladu co vidíš, to dostaneš. Princip tvorby www stránek těmito editory spočívá v tom, že autor pouze vkládá text a grafické objekty na plochu stránky a editor sám vygeneruje kód. Autor tedy nepotřebuje znát HTML jazyk. Ovšem výsledný HTML kód je pro následnou úpravu velmi nepřehledný a ve své podstatě již neumožňuje opravy.

Součástí WYSIWYG editorů je také standardní strukturní editor pomocí kterého lze provádět úpravu přímým vkládáním kódů.

### Strukturní editory

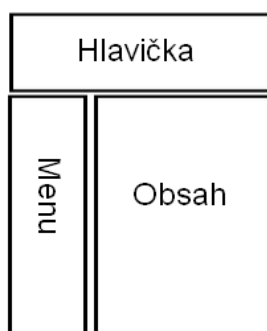
Strukturní editory jsou založeny na tvorbě stránek formou přímého vkládání kódů HTML jazyka. Z tohoto důvodu jsou vhodné zejména pro pokročilé autory www stránek, kteří znají samotný HTML jazyk. Výhodou je přehlednost a dobrá orientace ve výsledném HTML kódu.

## 6.3 Návrh vlastní elektronické publikace

Při seznámení se s elektronickými publikacemi jsem si pro návrh své elektronické publikace stanovil tyto priority:

- vizuálně jednoduché,
- přehledné menu,
- co největší velikost okna obsahu.

Přestože se tvorba www stránek pomocí rámců považuje za zastaralý způsob vytváření www stránek a příliš se nedoporučuje, tak v případě výukových publikací považuji tento způsob za dostačující. Proto jsem si pro návrh své vlastní elektronické publikace zvolil HTML dokument s rámy. Rozvržení celkem tří rámců je standardní. „Hlavička“ je ve formě horního pruhu v celé šířce stránky, „Menu“ je vlevo jako svislý pruh a „Obsah“ jako hlavní okno zabírá zbývající plochu.

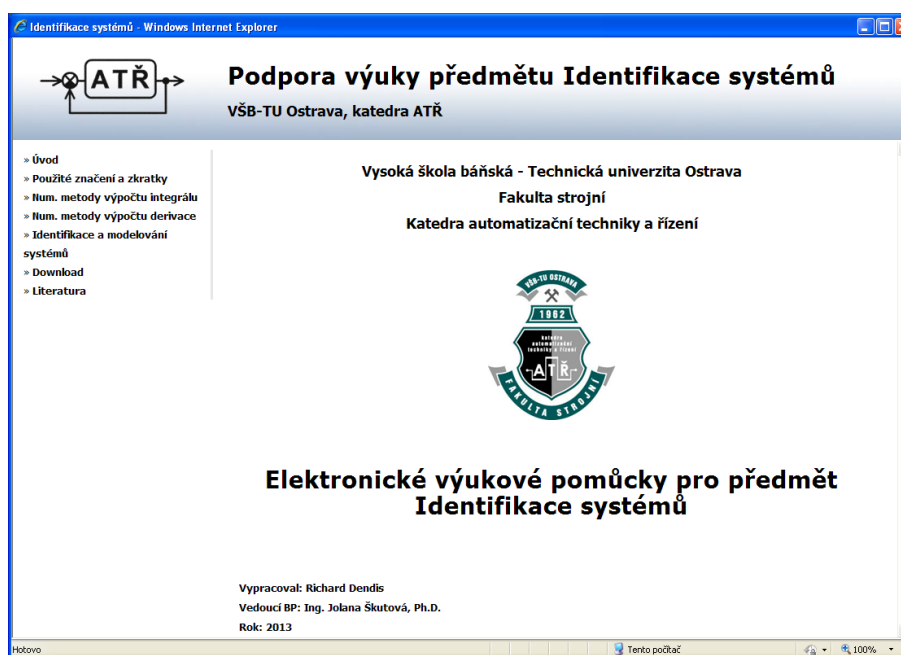


Obrázek 6.4 Navržené rozložení rámců

Celé publikaci dominuje na pozadí bílá barva, pouze pozadí hlavičky je podbarveno do světlého odstínu modré. Text je standardně černý. Za cíl jsem si kladl to, aby přílišná barevnost a sytost barev neodrazovala pozornost uživatelů od samotného obsahu.

Tematický obsah webových stránek ve formě elektronických výukových podkladů k předmětu Identifikace systémů koresponduje s obsahem bakalářské práce

- Numerické metody výpočtu integrálu.
- Numerické metody výpočtu derivace.
- Identifikace a modelování systémů.



Obrázek 6.5 Návrh vlastní elektronické publikace

V případě menu lze mít standardně rozevřenou jen jednu kapitolu. Aktivací názvu kapitoly dojde k rozevření nabídky podkapitol, následným klikem se načte obsah kapitoly, popř. se načte přímo podkapitola.

## 7 Závěr a zhodnocení dosažených výsledků

Cílem bakalářské práce bylo navrhnout a vytvořit elektronické výukové podklady k předmětu Identifikace systému.

V úvodních dvou kapitolách jsem se seznámil se způsoby řešení výpočtu integrálů a diferenciálních rovnic pomocí numerických metod. Součástí jednotlivých kapitol jsou příklady řešené pomocí skriptů vytvořeným v programu MATLAB. Zdrojové kódy k výpočtu jednotlivých numerických metod jsou vytvořeny v programu MATLAB jako m-soubory. Vybrané metody jsou v textové formě součástí bakalářské práce jako příloha. Ke stažení jsou k dispozici na DVD nosiči, nebo v rámci webových podkladů.

Obsahem třetí kapitoly se zaměřením na analytickou identifikaci je popis základních pojmů týkající se systému, jeho modelování a identifikaci. Vysvětleny jsou způsoby zápisu matematického modelu. V druhé části kapitoly se zaměřuji na postupy matematického řešení převodu vnějšího popisu dynamického systému na vnitřní stavový popis pomocí metod postupné integrace a snižování řádu derivace.

Čtvrtá kapitola je praktičtější a po úvodním seznámení se s programem MATLAB & Simulink sestavuji v Simulinku bloková schémata simulačního modelu k vzorovému příkladu pomocí metod postupné integrace a snižování řádu derivace. V následující páté kapitole vytvářím, pomocí programu Adobe Captivate, pro vzniklé simulační modely interaktivní prezentace jejich postupného sestavení.

V závěrečné šesté kapitole pokládám základy webových stránek, které jsou hlavním výstupem bakalářské práce. Navrhuji jejich koncept, vzhled a funkčnost. Výsledné webové stránky jsou určeny jako výukové podklady předmětu Identifikace systémů a jsou samostatnou přílohou bakalářské práce na DVD nosiči.

## Seznam použité literatury

CHYTIL, V. *Prezentační a výukový modul v prostředí Intranet/Internet pro analýzu a syntézu pomocí Delta transformace* [online]. Ostrava: VŠB-TU Ostrava, 2002. [cit. 2011-12-20]. Bakalářská práce, vedoucí: Vítěčková, M. Dostupný z [www: <http://books.fs.vsb.cz/DeltaTransAS/index.htm>](http://books.fs.vsb.cz/DeltaTransAS/index.htm).

DOSTÁL, P., GAZDOŠ, F. *Řízení technologických procesů*. 1. vyd. Zlín: FAI UTB ve Zlíně, 2006. 98 s. ISBN 80-7318-465-6.

DUŠEK, F. *Matlab a Simulink úvod do používání*. Pardubice: Univerzita Pardubice, 2000. 146 s. ISBN 80-7194-273-1.

FAJMON, B., RŮŽIČKOVÁ, I. *Matematika 3* [online].. Brno: VUT Brno, 2005. Dostupné z [www: <http://www.umat.feec.vutbr.cz/~hlavicka/skripta/matematika3.pdf>](http://www.umat.feec.vutbr.cz/~hlavicka/skripta/matematika3.pdf).

JANOVSKÝ, D. *Jak psát web* [online]. 2008. Dostupné z [www: <http://www.jakpsatweb.cz/>](http://www.jakpsatweb.cz/). ISSN 1801-0458.

KARBAN, P. *Výpočty a simulace v programech Matlab a Simulink*. Brno: Computer Press, 2006. 220 s. ISBN 978-80-251-1448-3.

KOLEKTIV AUTORŮ. *Vytváříme WWW stránky a spravujeme moderní web site*. Brno: Computer Press, 1998. ISBN 80-7226-080-4.

KUČERA, R. *Numerické metody*. Ostrava: VŠB-TU Ostrava, 2007. 152 s. ISBN 80-248-1198-7.

KUPKA, L. *Matlab & Simulink úvod do používání*. 1. vyd. Lanškroun: SOŠ a SOU Lanškroun, 2007. 167 s. ISBN 978-80-239-8871-0.

*MATLAB, Simulink* [online]. HUMUSOFT s.r.o. [cit. 2012-01-20]. Dostupné z [www: <http://www.humusoft.cz/produkty/matlab/matlab/>](http://www.humusoft.cz/produkty/matlab/matlab/).

MINÁŘ, K. *Analýza lineárních regulačních obvodů* [online]. Ostrava: VŠB-TU Ostrava, 2000. [cit. 2011-12-20]. Bakalářská práce, vedoucí: Wagnerová, R. Dostupný z [www: <http://books.fs.vsb.cz/Analyza/index.html>](http://books.fs.vsb.cz/Analyza/index.html).

MINÁŘ, M. *Prezentační a výukový modul v prostředí Intranetu pro oblast syntézy regulačních obvodů* [online]. Ostrava: VŠB-TU Ostrava, 2000. [cit. 2011-12-20]. Bakalářská práce, vedoucí: Wagnerová, R. Dostupné z www: <[http://www.352.vsb.cz/uc\\_texty/Synteza/index.htm](http://www.352.vsb.cz/uc_texty/Synteza/index.htm)>.

NAVARA, M., NĚMEČEK, A. *Numerické metody* [online]. Praha: ČVUT Praha, 2001. Dostupné z www: <<http://cmp.felk.cvut.cz/~navara/nm/NM.pdf>>.

NĚMEC, L. *Adobe Captivate 5: e-learning na úrovni* [online]. Publikováno 12.07.2010 [cit. 2012-01-20]. Dostupné z www: <<http://www.grafika.cz/rubriky/software/adobe-captivate-5-e-learning-na-urovni-137868cz>>.

NOSKIEVIČ, P. *Modelování a identifikace systémů*. Ostrava: Montanex, 2007. 276 s. ISBN 80-7225-030-2.

NOSKIEVIČ, P. *Simulace systémů*. 1. vyd. Ostrava: VŠB-TU Ostrava, 1992. 217 s. ISBN 80-7078-112-2.

*Using Adobe Captivate 5.0 & 5.5* [online] [cit. 2012-01-20]. Dostupný z www: <[http://help.adobe.com/en\\_US/captivate/cp/using/index.html](http://help.adobe.com/en_US/captivate/cp/using/index.html)>.

VÍTEČKOVÁ, M., VÍTEČEK, A. *Základy automatické regulace*. 2. vyd. Ostrava: VŠB-TU Ostrava, 2008. 244 s. ISBN 978-80-248-1924-2.



## Seznam příloh

**Příloha A – Numerická integrace (výpis zdrojového kódu)**

**Příloha B – Numerická derivace (výpis zdrojového kódu)**

**DVD médium:**

- Zadání bakalářské práce
- Vypracovaná bakalářská práce
- Záznam o závěrečné práci
- Webové stránky: Elektronické výukové pomůcky pro předmět Identifikace systémů

## Příloha A – Numerická integrace (výpis zdrojového kódu)

### Obdélníková metoda

```
function I_obd=int_obd(popisFcn, a, b, n)
% int_obd - vypocet urcitého integrálu pomocí obdelnikoveho pravidla
%
% Vstupy
%   popisFcn   nazev funkce
%   a          dolni mez
%   b          horni mez
%   n          pocet dilku
%
% Vzorove spusteni souboru
%   Nadefinovat vstupni promenne
%   Zapsat rovnici funkce v Command Window (napr.
popisFcn=inline('sin(x)'))
%   I_obd=int_obd(popisFcn, a, b, n)
%
% Vystup
%   I_obd      priblizna hodnota integralu

if a>b
    error('Chybne zadane krajni body intervalu!')
end

if n<1
    error('Pocet dilku musi byt kladny!')
end

h = (b-a)/n; % krok
I_obd = 0; % pocatecni velikost integralu

for i=a:h:b-h
    y = feval(popisFcn,i+h/2); % vypocet funkcní hodnoty v bode 'i'
    I_obd = I_obd + h*y; % pridani 1 obdelniku
end
```

### Lichoběžníková metoda

```
function I_lich=int_lich(popisFcn, a, b, n)
% int_lich - vypocet urcitého integrálu pomocí lichobežnikoveho pravidla
%
% Vstupy
%   popisFcn   nazev funkce
%   a          dolni mez
%   b          horni mez
%   n          pocet dilku
%
% Vzorove spusteni souboru
%   Nadefinovat vstupni promenne
```

```
%      Zapsat rovnici funkce v Command Window (napr.
popisFcn=inline('sin(x)'))
%      I_lich=int_lich(popisFcn, a, b, n)
%
%      Vystup
%      I_lich      priblizna hodnota integralu

if a>=b
    error('Chybne zadane krajni body intervalu!')
end

if n<1
    error('Pocet dilku musi byt kladny!')
end

h=(b-a)/n; % krok
x=a:h:b;

fx=feval(popisFcn,x);          % vypocet funkcni hodnoty v bode 'x'

I_lich=h/2*(fx(1)+2*sum(fx(2:n))+fx(n+1));
```

### Simpsonova metoda

```
function I_simpson=int_simp(popisFcn,a,b,n)
% int_simp - vypocet urciteho integralu pomoci Simpsonova pravidla
%
%      Vstupy
%      popisFcn    nazev funkce
%      a           dolni mez
%      b           horni mez
%      n           pocet dilku
%
%      Vzorove spusteni souboru
%      Nadefinovat vstupni promenne
%      Zapsat rovnici funkce v Command Window (napr.
popisFcn=inline('sin(x)'))
%      I_simpson=int_simp(popisFcn,a,b,n)
%
%      Vystup
%      I_simpson   priblizna hodnota integralu

if a>b
    error('Chybne zadane krajni body intervalu!')
end

h=(b-a)/n; % krok
x=a:h:b;

fx=feval(popisFcn,x);          % vypocet funkcni hodnoty v bode 'x'

I_simpson=h/3*(fx(1)+4*sum(fx(2:2:n))+2*sum(fx(3:2:n-1))+fx(n+1));
```

## Příloha B – Numerická derivace (výpis zdrojového kódu)

### Eulerova metoda

```
function [xAprox,yAprox]=EulerovaMetoda (popisFcn,xmin,xmax,h,y0)
% EulerovaMetoda - metoda numerickeho reseni diferencialni rovnice
% y'=f(x,y) s danou pocatecni podminkou y(0)=y0
%
% Vstupy
% popisFcn   nazev funkce
% xmin       dolni mez intervalu
% xmax       horni mez intervalu
% h          delka kroku vypoctu
% y0         pocatecni podminka
%
% Vzorove spusteni souboru
% Nadefinovat vstupni promenne
% Zapsat rovnici funkce (napr. f=inline('x*y+x*exp(x.^2)'))
% [xAprox,yAprox]=EulerovaMetoda (popisFcn,xmin,xmax,h,y0)
%
% Vystupy
% xAprox     vektor usecky
% yAprox     vektor souradnic

pocet=(xmax-xmin)/h;           %stanoveni poctu reseni
xAprox(1,1)=xmin;
yAprox(1,1)=y0;

for k=1:pocet;
    x=xAprox(k,1);
    y=yAprox(k,1);
    f=feval (popisFcn, x, y);   % vypocet funkcní hodnoty v bode 'x' a 'y'
    yAprox(k+1,1)=yAprox(k,1)+h*f;
    xAprox(k+1,1)=xAprox(k,1)+h;
end
```

### Runge-Kuttova metoda 2.řádu

```
function [xAprox,yAprox]=RungeKutt2radu (popisFcn,xmin,xmax,h,y0)
% RungeKutt2radu - metoda numerickeho reseni diferencialni rovnice
% y'=f(x,y) s danou pocatecni podminkou y(0)=y0
% vyuzivajici k vypoctu clenů vyšších řádů
%
% Vstupy
% popisFcn   nazev funkce
% xmin       dolni mez intervalu
% xmax       horni mez intervalu
% h          delka kroku vypoctu
% y0         pocatecni podminka
%
% Vzorove spusteni souboru
% Nadefinovat vstupni promenne
% Zapsat rovnici funkce v Command Window (napr.
popisFcn=inline('x*y+x*exp(x.^2)'))
% [xAprox,yAprox]=RungeKutt2radu (popisFcn,xmin,xmax,h,y0)
%
% Vystupy
```

```

%   xAprox      vektor usecky
%   yAprox      vektor souradnic

pocet=(xmax-xmin)/h;           % stanoveni poctu reseni
xAprox(1,1)=xmin;
yAprox(1,1)=y0;

for k=1:pocet
    x=xAprox(k,1);
    y=yAprox(k,1);
    k1=feval(popisFcn,x,y); % vypocet 1. pomocneho bodu

    x=xAprox(k,1)+h/2;
    y=yAprox(k,1)+h/2*k1;
    k2=feval(popisFcn,x,y); % vypocet 2. pomocneho bodu

    yAprox(k+1,1)=yAprox(k,1)+h*k2;
    xAprox(k+1,1)=xAprox(k,1)+h;
end

```

### Runge-Kuttova metoda 4.řádu

```

function [xAprox,yAprox]=RungeKutt4radu(popisFcn,xmin,xmax,h,y0)
% RungeKutt4radu - metoda numerickeho reseni diferencialni rovnice
%                  y'=f(x,y) s danou pocatecni podminkou y(0)=y0
%                  vyuzivajici k vypoctu clen vyssich radu

% Vstupy
%   popisFcn     nalez funkce
%   xmin         dolni mez intervalu
%   xmax         horni mez intervalu
%   h            delka kroku vypoctu
%   y0           pocatecni podminka
%
% Vzorove spusteni souboru
%   Nadefinovat vstupni promenne
%   Zapsat rovnici funkce v Command Window (napr.
popisFcn=inline('x*y+x*exp(x.^2)')
%   [xAprox,yAprox]=RungeKutt4radu(popisFcn,xmin,xmax,h,y0)
%
% Vystupy
%   xAprox      vektor usecky
%   yAprox      vektor souradnic

pocet=(xmax-xmin)/h;           % stanoveni poctu reseni
xAprox(1,1)=xmin;
yAprox(1,1)=y0;

for k=1:pocet
    x=xAprox(k,1);
    y=yAprox(k,1);
    k1=feval(popisFcn, x, y); % vypocet 1. pomocneho bodu

    x=xAprox(k,1)+h/2;
    y=yAprox(k,1)+h/2*k1;
    k2=feval(popisFcn, x, y); % vypocet 2. pomocneho bodu

    x=xAprox(k,1)+h/2;
    y=yAprox(k,1)+h/2*k2;

```

```

    k3=feval (popisFcn, x, y);    % vypocet 3. pomocneho bodu

    x=xAprox (k,1)+h/2;
    y=yAprox (k,1)+h/2*k3;
    k4=feval (popisFcn, x, y);    % vypocet 4. pomocneho bodu

    yAprox (k+1,1)=yAprox (k,1)+h*k4;
    xAprox (k+1,1)=xAprox (k,1)+h;

end

```

### Adams-Bashforthova metoda (4-kroková)

```

function [xAprox,yAprox]=AdamsBashf2krok (popisFcn,xmin,xmax,h,y0,y1)
% AdamsBashf2krok - metoda numerickeho reseni diferencialni rovnice
% y'=f(x,y) s danou pocatecni podminkou y(0)=y0
% vychazejici z implicitnich vicekrokovych metod

% Vstupy
% popisFcn   nazev funkce
% xmin       dolni mez intervalu
% xmax       horni mez intervalu
% h          delka kroku vypoctu
% y0         pocatecni podminka
% y1         hodnota derivace pro i=1 (k vypoctu pouzit
RungeKutt4radu.m)
%
% Vzorove spusteni souboru
%   Nadefinovat vstupni promenne
%   Zapsat rovnici funkce v Command Window (napr.
popisFcn=inline('x*y+x*exp(x.^2)')
%   [xAprox,yAprox]=AdamsBashf2krok (popisFcn,xmin,xmax,h,y0,y1)
%
% Vystupy
% xAprox     vektor usecky
% yAprox     vektor souradnic

pocet=(xmax-xmin)/h;    % stanoveni poctu reseni
xAprox (1,1)=xmin;
yAprox (1,1)=y0;

xAprox (2,1)=xmin+h;
yAprox (2,1)=y1;

for k=2:pocet,
    x=xAprox (k-1,1);
    y=yAprox (k-1,1);
    k2=feval (popisFcn,x,y);

    x=xAprox (k,1);
    y=yAprox (k,1);
    k1=feval (popisFcn,x,y);

    yAprox (k+1,1)=yAprox (k,1)+h*(1.5*k1-0.5*k2);
    xAprox (k+1,1)=xAprox (k,1)+h;

end

```